ISSN 0542 9951

2
1996

NAJSTARSZE PISMO INFORMATYCZNE W POLSCE – UKAZUJE SIĘ OD 1965 ROKU

informatyka

POLITECHNIKA ŚLĄSKA Biblioteka Główna Czytelnia
P 1877/96

miesięcznik profesjonalistów

Integracja i partnerstwo

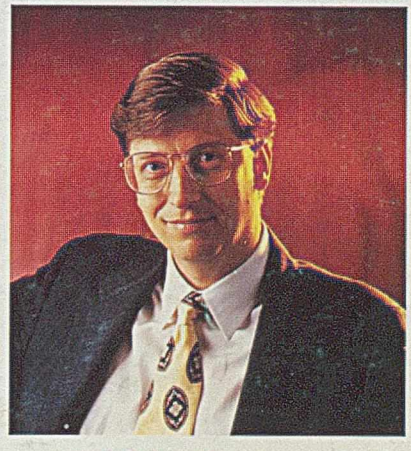
Każdego roku obiecuję sobie, że już nigdy więcej nie pojawię się w tym zbitym tłumie ludzi pędzących nie wiadomo za czym, nie wiadomo gdzie. Najgorsza jest wirtualna rzeczywistość. Wszyscy dookoła zachowują się jakby sami mieli hełmy na głowach, a przed oczami kokpit Jumbo Jeta. Korek absolutny, już jestem bliski zrezygnowania z przebicia się na drugą stronę. Łokciowanie wymaga ode mnie zaparcia się pewnych zasad, które wpojono mi w zamierzchłym dzieciństwie. Uff, udało się. Jestem po drugiej stronie.

Osiągnąwszy tak oczywisty sukces ustaję na chwilę w biegu i zaczynam się chłodno zastanawiać, jaki jest właściwie jego cel. Po pierwsze, przyjąłem oczywiste założenie, że każdy wystawca bez względu na wielkość ma równe szanse. Po drugie, postanowiłem chwilę zastanowić się, czym różniły się bieżące targi od tych z poprzedniego roku.

Moją wyczuloną uwagę zwróciło kilka spraw. Zatem po kolei. Już stało się tradycją, że wysocy funkcjonariusze rządowi otwierają targi podkreślając w ten sposób rangę informatyki w Polsce. Tak też było tym razem. Wicepremier Grzegorz Kołodko przekraczając progi hali wystawowej na Mokotowie nie miał wątpliwości, że: *Komputeryzacja wkroczyła nieodwołalnie niemal do każdej dziedziny życia gospodarczego i społecznego i nie ma przed nią odwrotu. Sama w sobie jest jednym z istotnych wyznaczników i wskaźników poziomu ekonomicznego, stymuluje rozwój całej gospodarki.* Tak, to prawda. Ciesząc się z faktu, że coraz większa liczba ludzi w sferach rządowych dostrzega konieczność rozwijania i stosowania narzędzi komputerowych w gospodarce, trzeba dodać, że na obecnym etapie rozwoju naszego kraju chodzi już o coś więcej - stosowanie środków informatyki, czyli całego zespołu zasobów komputerowych w taki sposób, aby nie tylko zbudować działający system informatyczny, ale również przekształcić w od-

dokończenie na s. 5

Fragment ekspozycji targowej nagrodzonej przez dziennikarzy „Złotą myszą"



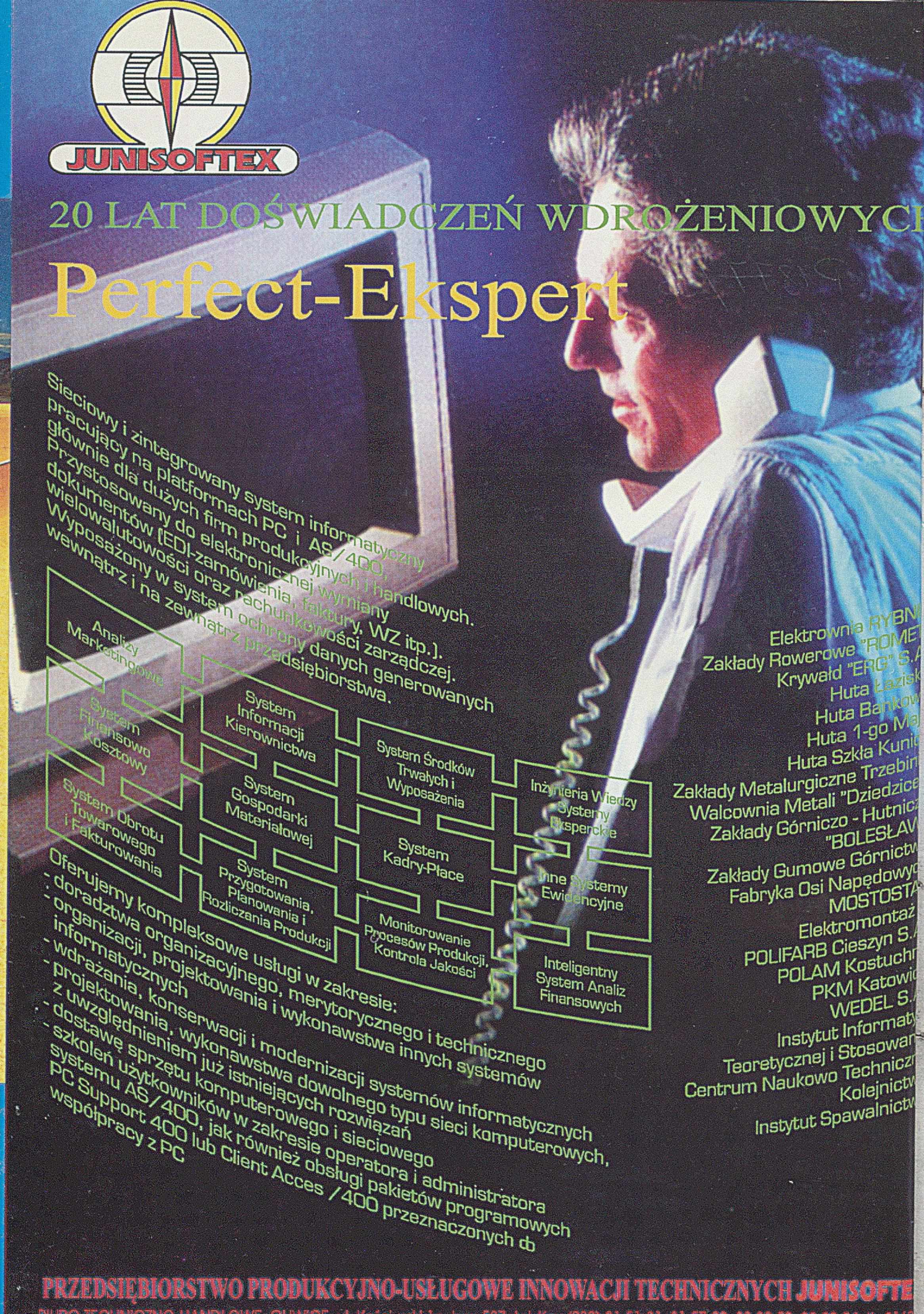
Bill Gates – przypadek spełnionego marzenia

Chyba wszystko już co możliwe napisano na temat Billa Gatesa i jego wizyty w Polsce. Nie mogę jednak oprzeć się chęci podzielenia się z Czytelnikami pewnymi myślami, które mi się nasunęły w czasie spotkań z szefem Microsoftu. Nazywają go wizjonerem, on sam zaczyna opowieści o oszałamiającym sukcesie, jaki stał się udziałem jego i Microsoftu od powiedzenia, że jest przypadkiem spełnionego marzenia. Mówi też, że znalazł się we właściwym miejscu o właściwym czasie. Warto zwrócić uwagę na to, gdzie Gates „odkrył" swoje powołanie, czyli jak sam doszedł do tego, iż jego największą przyjemnością jest tworzenie programów, które staną się cząstką życia milionów ludzi. Otóż zaczęło to się w 1968 r. w liceum, którego dyrekcja doszła do wniosku, że należałoby zapoznać uczniów z informatyką. Nie było wtedy jeszcze PC-tów, ale Klub Mamuś, odpowiednik naszego komitetu rodzicielskiego, znalazł środki na kupno czasu obliczeniowego od firmy posiadającej duży komputer, z którym można było kontaktować się za pomocą terminali. Tam to właśnie, Gates ucząc się pisania programów w BASICU-u, odkrył w sobie trwają-

dokończenie na s. 2

WYDAWNICTWO SIGMA NOT

Cena 5,00 zł (50 000 zł)


JUNISOFTEX
20 LAT DOŚWIADCZEŃ WDROŻENIOWYCH
Perfect-Ekspert
Sieciowy i zintegrowany system informatyczny pracujący na platformach PC i AS/400, przystosowany do elektronicznej wymiany dokumentów (EDI-zamówienia, faktury, WZ itp.). Wyposażony w system ochrony danych generowanych wewnątrz i na zewnątrz przedsiębiorstwa.
głównie dla dużych firm produkcyjnych i handlowych.
wielowalutowości oraz rachunkowości zarządczej.
Analizy Marketingowe
System Finansowo Kosztowy
System Informacji Kierownictwa
System Środków Trwałych i Wyposażenia
System Gospodarki Materiałowej
System Kadry-Płace
System Obrotu Towarowego i Fakturowania
System Przygotowania, Planowania i Rozliczania Produkcji
Monitorowanie Procesów Produkcji, Kontrola Jakości
Inżynieria Wiedzy Systemy Eksperckie
Inne Systemy Ewidencyjne
Inteligentny System Analiz Finansowych
Oferujemy kompleksowe usługi w zakresie:
- doradztwa organizacyjnego, merytorycznego i technicznego
- organizacji, projektowania i wykonawstwa innych systemów informatycznych
- wdrażania, konserwacji i modernizacji systemów informatycznych
- projektowania, wykonawstwa dowolnego typu sieci komputerowych, z uwzględnieniem już istniejących rozwiązań komputerowego i sieciowego
- dostawę sprzętu komputerowego i sieciowego
- szkoleń użytkowników w zakresie operatora i administratora systemu AS/400, jak również obsługi pakietów programowych PC Support 400 lub Client Acces /400 przeznaczonych do współpracy z PC
Elektrownia RYBN
Zakłady Rowerowe "ROME
Krywałd "ERG" S.A
Huta Łazisk
Huta Bankow
Huta 1-go Ma
Huta Szkła Kunic
Zakłady Metalurgiczne Trzebin
Walcownia Metali "Dziedzice
Zakłady Górniczo - Hutnicz
"BOLESŁAV
Zakłady Gumowe Górnictw
Fabryka Osi Napędowyc
MOSTOSTA
Elektromontaż
POLIFARB Cieszyn S.A
POLAM Kostuchn
PKM Katowic
WEDEL S.A
Instytut Informaty
Teoretycznej i Stosowan
Centrum Naukowo Technicz
Kolejnictw
Instytut Spawalnictw
PRZEDSIĘBIORSTWO PRODUKCYJNO-USŁUGOWE INNOWACJI TECHNICZNYCH JUNISOFTE

informacje

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P1877/96

Konkursy, nagrody, laureaci

W czasie Targów Komputer-Expo, podobnie jak w minionych latach, wręczano nagrody w różnego rodzaju konkursach związanych z branżą informatyczną.

W mającym najdłuższe tradycje konkursie INFOSTAR '96 laureatami zostali:

– **W kategorii rozwiązań informatycznych – Wiesław Paluszyński** za profesjonalny rozwój Systemów Informatycznych Monitorowania Środowiska w Ministerstwie Ochrony Środowiska i Zasobów Naturalnych. Laureat jest zastępcą Głównego Inspektora Ochrony Środowiska i pełnomocnikiem ministra ds. informatyki. Członek prezydium Rady Koordynacyjnej ds. Teleinformatyki przy RM URM. Absolwent Wydz. Elektroniki Politechniki Warszawskiej w 1984 r.;

– **W kategorii osiągnięć biznesowych – Tomasz Sielicki** – za rozwój i przekształcenie spółki Computerland w spółkę akcyjną oraz wprowadzenie jej na giełdę. Prezes zarządu Computerland Poland S.A., następnie jej dyrektor generalny. Do 1985 r. był pracownikiem techniczno-naukowym w Instytucie Informatyki, następnie pracował w przedsiębiorstwie Karen. Absolwent Instytutu Informatyki Wydz. Elektroniki Politechniki Warszawskiej w 1983 r.;

– **W kategorii propagowania Informatyki – Jerzy Nowak** – za ... propagowanie informatyki. Organizator i szef Centrum Informatyki w Zakładach Mechanicznych Bumar Łabędy S.A.. Członek i założyciel Polskiego Towarzystwa Informatycznego i jego oddziału w Katowicach. Obecnie sekretarz generalny PTI, organizator Szkoły PTI w Szczyrku i Forum Technologii Informatycznych w Mrągowie. Absolwent Wydz. Mechaniczno-Technologicznego Politechniki Warszawskiej w specjalizacji: organizacja produkcji i zarządzanie. Ukończył też dwuletnie studia podyplomowe z zakresu informatyki;

Nagrody Prezydenta Krajowej Izby Gospodarczej otrzymali:

– w kategorii firma polska **INWAR S.A.** z Sieradza za dużą dynamikę sprzedaży, uzyskanie międzynarodowej normy jakości ISO 9002 oraz rozwój eksportu do krajów skandynawskich;

– w kategorii firm zagranicznych – **ORACLE Polska** – za rozwój sieci polskich firm partnerskich produkujących aplikacje;

– w kategorii „człowiek" – **red. Zbigniew Zwierzchowski** z „Rzeczpospolitej" – za rzetelną popularyzację informatyki w mediach skierowanych do najszerszej publiczności.

Dziennikarze akredytowani przy targach Komputer Expo '96 w plebiscycie na najlepszą ekspozycję targową przyznali **„Złotą mysz" firmie IBM.**

(kar).

Stowarzyszenie Użytkowników Systemów Inżynierskich

W grudniu 1995 r. zarejestrowano **Stowarzyszenie Użytkowników Systemów Inżynierskich – SUSI.** Przez systemy inżynierskie założyciele rozumieją kompleks spraw związanych ze sprzętem, oprogramowaniem, bazami danych, urządzeniami oraz systemami łączności, automatyki i sterowania niezbędnych dla automatyzacji procesu projektowania, produkcji i zarządzania. Celem Stowarzyszenia jest m.in. popularyzowanie metod planowania, projektowania, uruchamiania, wdrożeń i zastosowań informatycznych systemów inżynierskich, działania na rzecz zgodności norm i standardów , konsultacje, inicjowanie działań legislacyjnych i normalizacyjnych, wspieranie różnego rodzaju działalności edukacyjnej, naukowej, technicznej i normalizacyjnej. Do SUSI przystąpiło około 100 członków reprezentujących zakłady przemysłowe, samorząd, firmy produkujące lub dostarczające systemy inżynierskie. Na I Walnym Zebraniu Członków SUSI, które odbyło się 22.01. 1996 r. w Warszawie dokonano wyboru władz. Prezesem Stowarzyszenia został jego inicjator i założyciel **Bogusław Dziarnowski** – Pełnomocnik Ministra Przemysłu i Handlu oraz wiceprzewodniczący Rady Koordynacyjnej ds. Teleinformatyki przy Radzie Ministrów RP. Zainteresowani działalnością SUSI mogą uzyskać informacje pod adresem: SUSI, 01-793 Warszawa, ul. Rydygiera 8, tel/faks 639-86-11, 629-42-27, e-mail: boguslaw.dziarnowski@mit.gov.pl.

ATI w Polsce

W czasie trwania targów Komputer Expo 96 odbyła się w Warszawie konferencja prasowa zorganizowana przez firmę **ATI** *(Allied Telesyn International)*, obecną już od pewnego czasu na naszym rynku, i firmę **ASCOMP**, jej integratora systemowego na Polskę.

Produktem istniejącej od 1987 r. amerykańskiej firmy ATI jest CentreCom - szeroka oferta podstawowych elementów budulcowych dla sieci lokalnych, opierająca się na nowoczesnej technologii i masowej produkcji: karty sieciowe, koncentratory, mosty (ang. *bridge*), przełączniki (ang. *switch*), sprzęgacze (ang. *transceiver*), wzmacniaki (ang. *repeater*) oraz odpowiednie oprogramowanie do zarządzania. Globalną strategią ATI, w sytuacji nasycenia rynku światowego produktami różnych firm o zbliżonej klasie, jest zdobywanie klientów przez stosowanie najnowszej technologii, pełną zgodność z istniejącymi standardami, sprawną obsługę klientów i - dzięki masowej produkcji - konkurencyjne ceny. ATI rozprowadza swoje produkty przez sieć dystrybutorów, nie prowadząc samodzielnej sprzedaży detalicznej. Według oceny ATI, Polska jest dobrym, bezpiecznym rynkiem o dużej dynamice, czego dowodzi wydzielenie go przez ATI jako odrębnej strefy działalności. (pb)

To nie sen to Java

Pod tym hasłem w przeddzień targów Komputer Expo-96, odbyło się niezwykle interesujące seminarium firmy Sun Microsystems poświęcone najnowszym narzędziom i oprzyrządowaniu umożliwiającym eksplorację i wykorzystywanie sieci Internet, w tym również do celów komercyjnych. Po wysłuchaniu prezentacji dotyczących nowo powstałego języka Java i napisanej w tym języku przeglądarki dla sieci Internet o nazwie Hot Java, nie można mieć wątpliwości, że głoszone ostatnio proroctwa Larry Ellisona, szefa firmy Oracle, nie są żadną fantazją. Język Java i przeglądarka Hot Java są pierwszym krokiem do szybkiego urzeczywistnienia wizji bardzo tanich, ale silnych terminali sieciowych udostępnianych zwłaszcza użytkownikom telewizyjnych sieci kablowych w cenie 300-500 dolarów.

Obiektowy język Java wykazuje ogromne podobieństwo do znanych języków C i C++, dlatego też programista znający te języki będzie mógł natychmiast go wykorzystywać. Prawdziwa siła nowego języka kryje się w koncepcji środowiska, dla którego jest pisane oprogramowanie w Javie. Otóż programy nie są kompilowane na kod maszynowy, lecz na bajtowy kod pośredni, który jest interpretowany przez maszyny wirtualne działające na różnych procesorach. Dowolne programy zapisane w postaci kodu pośredniego będzie można wykonywać po uprzednim ściągnięciu ich za pośrednictwem sieci do odpowiedniego środowiska systemowego, a także do wspomnianych wcześniej terminali.

Pierwszym programem powstałym według tej idei, który ma szansę odnieść olbrzymi sukces, jest przeglądarka Hot Java dla sieci Internet. Umożliwia ona realizację tzw. appletów (to znaczy właśnie programów pisanych w Javie) ściąganych po sieci, podobnie jak do tej pory działo się z tekstami, dźwiękiem i obrazami. Applety zaś w ramach przeglądanych stron będą umożliwiać dowolne działania interakcyjne, np. dynamiczne oglądanie wyników giełdowych, symulowanie reakcji chemicznych, animację obrazów itp. O znaczeniu nowej przeglądarki Hot Java świadczy natychmiastowe zastosowanie tej idei w najpopularniejszym ostatnio produkcie tego typu Netscape Navigator v.2.0. **(JS)**

Bill Gates - przypadek spełnionego marzenia

dokończenie z I s. okładki

cą do dziś pasję do programowania, a także zdolności do interesów. Nie bez powodu przypominam ten epizod z życia twórcy MS DOS i Windows. Wniosek prosty: wspierajmy wszelkimi sposobami te komitety rodzicielskie i dyrekcje polskich szkół, które dokonując przeróżnych sztuczek, zdobywając cudem sponsorów, tworzą pracownie komputerowe i podłączają szkoły do Internetu.

Od 1975 r., kiedy to Gates z Allenem założyli Microsoft, a właściwie od 1981 r., kiedy to narodził się IBM PC i powstał MS DOS firmy Microsoft, Bill Gates nadal uparcie wciela w życie swoją wizję świata, w której komputer, obok telefonu i telewizora, znajdzie się w każdym domu. Oczywiście, ta wizja ciągle jest modyfikowana przez rozwój techniki i technologii. Dziś, co podkreślał Gates w każdym ze swoich licznych warszawskich wystąpień, tą szansą na dotarcie komputerów „pod strzechy" jest sieć komputerowa. Jej dalszy rozwój wymaga szybkich łączy światłowodowych, a tam, gdzie doprowadzenie światłowodów jest zbyt kosztowne, trzeba będzie używać łączności satelitarnej. I to jest ta druga ważna refleksja nasuwająca się, jak sądzę, wielu uczestnikom spotkań z Gatesem: bez powszechnie dostępnej i dobrej sieci telekomunikacyjnej nie ma szans na to, by komputery były w pełni wykorzystane, a tym samym na przepustkę do nowoczesności. Miejmy nadzieję, że dostrzegli ten aspekt wizyty Billa Gatesa w Polsce również politycy oraz ludzie odpowiedzialni za rozwój telekomunikacji.

W tym, że nie ma rozwoju informatyki bez dostępu do sieci, są z Billem Gates zgodni twórcy innego, konkurencyjnego spojrzenia na rozwój informatyki: idei Network Computer, czyli łatwego dostępu do zasobów sieci i korzystania z niej za pośrednictwem tanich komputerów, czegoś w rodzaju dawnych terminali. Gates natomiast uważa, że komputery stojące na naszych biurkach będą coraz mocniejsze i będą wymagały coraz bardziej rozbudowanych, ale i coraz prostszych w obsłudze programów, których oczywiście, dostarczy przede wszystkim Microsoft.

– *Zazdroszczę młodym ludziom, którzy rodzą się w świecie PC-tów i dostępu do Internetu, to ich przygotuje do pracy, niekoniecznie w informatyce, ale i w każdej innej dziedzinie* – stwierdził Bill Gates na konferencji prasowej. Dziś, gdy nabrzmiewają u nas problemy związane z opłatami za Internet, dostępnością, wysokością opłat, a także jakością połączeń telefonicznych, rodzi się pytanie: czas jest na pewno odpowiedni na realizację wizji Gatesa, ale czy miejsce jest właściwe? Mówiąc prościej, byłoby wspaniale, by ta wizyta uświadomiła ludziom, którzy u nas decydują o rozwoju telekomunikacji, że jest to także nasza gra o przyszłość.

Krystyna Karwicka

Solide Edge firmy Intergraph

Amerykańska firma Intergraph zaprezentowała na konferencji prasowej w styczniu w Warszawie pakiet Solid Edge – oprogramowanie CAD nowej generacji służące do modelowania części mechanicznych i tworzenia złożeń.

Solid Edge oferuje projektantom-mechanikom przestrzenne rozwiązanie CAD, które powstało w wyniku zastosowania najnowszej technologii Intergraphu o nazwie Jupiter. Dotąd projektanci byli zwykle zmuszani do wyboru pomiędzy najlepszymi rozwiązaniami w swojej klasie i pełną integracją, pomiędzy łatwością użytkowania i możliwościami, pomiędzy rozwiązaniami drogimi i tanimi. Dzięki wspieraniu standardów przemysłowych, wykorzystaniu silnego i opartego na intuicji środowiska Windows oraz korzystnej cenie, Solid Edge eliminuje te niedogodności oferując pierwszy przemysłowy system zorientowany na projektowanie zespołów. „Solid Edge reprezentuje nową koncepcję oprogramowania mechanicznego opartą na otwartej, środowiskowej współpracy realizowanej przez Microsoft i Intergraph", powiedział Bruce Jenkins – wiceprezydent Daratech. „Każdy poważny użytkownik systemów CAD powinien zwrócić uwagę na Solid Edge".

Solid Edge różni się znacząco od innych systemów CAD niespotykaną łatwością użytkowania, unikalnym podejściem do projektowania zespołów i wykorzystaniem technik OLE oraz „przeciągnij i upuść" dla danych typu CAD. Solid Edge, stworzony specjalnie dla Windows 95 i Windows NT, jest pierwszym opartym na środowisku Windows mechanicznym systemem CAD; zaimplementowano w nim innowacyjne funkcje projektowania zespołów i rewolucyjną technologię integracji oprogramowania. Ta przewaga technologiczna umożliwia skrócenie czasu wejścia produktu na rynek i poprawienie wydajności projektowania.

Solid Edge wykorzystuje także wielkie możliwości modelowania bryłowego ACIS stworzonego przez Spatial Technology. ACIS zapewnia szybkość, stabilność i niezawodne mechanizmy wymiany danych z innymi aplikacjami opartymi na tym standardzie.

(JS)

Microsoft Works 4.0

Doczekaliśmy się kolejnej polskiej aplikacji dla Windows 95. 16 stycznia 1996 firma Microsoft zaprezentowała polską wersję popularnego pakietu zintegrowanego Microsoft Works dla Windows 95. Para prezenterów z firmy Microsoft zademonstrowała duże możliwości tego znacznie tańszego pakietu od MS Office w interesujący i wciągający słuchaczy sposób.

Pakiet Microsoft Works 4.0 składa się z kilku zintegrowanych ze sobą modułów – edytora tekstów, arkusza obliczeniowego, programu do tworzenia baz danych, programu graficznego oraz komunikacyjnego. Wysoka skala zintegrowania pakietu ze środowiskiem systemów Windows pozwala na niezwykle proste przemieszczanie różnego typu obiektów – tekstów, tabel, rekordów i obrazów – z modułu do modułu. Dzięki zastosowaniu tzw. kreatorów, które umożliwiają konfigurowanie zarówno samych modułów, jak i opracowywanych przy ich pomocy dokumentów, praca z pakietem Works jest bardzo prosta nawet dla komputerowych nowicjuszy.

Pakiet Microsoft Works 4.0 wyposażono w wiele nowych właściwości; poniżej kilka ważniejszych:

- jest aplikacją 32-bitową, wykorzystującą ogromne możliwości i wydajność systemu operacyjnego Windows 95;
- posiada menu Uruchamiania Zadań, umożliwiające natychmiastowe przystąpienie do wykonania właściwego zadania;
- uwidacznia Książkę Adresową na pasku narzędzi, zapewniając dostęp do ważnych informacji nt. osób i firm;
- umożliwia wykorzystywanie różnych obiektów przy pomocy techniki OLE z poziomu paska menu;
- zawiera edytor równań pozwalający na stosowanie symboli i formuł matematycznych;
- zapewnia realizowanie połączeń telefonicznych, faksowych i poczty elektronicznej, jeśli istnieje odpowiednie oprogramowanie i osprzęt komunikacyjny. **(JS)**

RODAN SYSTEM wygrał przetarg MPiPS

Firma RODAN SYSTEM wygrała zamknięty przetarg i podpisała kontrakt z Ministerstwem Pracy i Polityki Socjalnej na realizację Pilotowego Systemu Zarządzania Dokumentami i Przebiegiem Pracy. Przedsięwzięcie jest finansowane z funduszy PHARE, wartość kontraktu wynosi ok. 500 tys. ECU. RODAN był jedyną polską firmą zaproszoną do przetargu. System ma być oparty na zrealizowanym w firmie środowisku automatyzacji prac biurowych, którego architektura jest efektem integracji trzech produktów: IBM FlowMark (funkcje zarządzania przepływem prac), Lotus Notes (funkcje zarządzania dokumentami) oraz Office Objects realizującego funkcje przetwarzania obrazów dokumentów oraz zarządzania systemem hierarchicznej pamięci służącej do przechowywania dokumentów. System Office Objects został zrealizowany w RODAN System jako biblioteka klas obiektów, zgodna ze specyfikacją Object Management Group, a funkcje zarządzania hierarchiczną pamięcią zrealizowano opierając się na mechanizmach aktywnej bazy danych CA-OpenINGRES. Projekt systemu zostanie wykonany na podstawie metodyki Object Modeling Technique (OMT), z wykorzystaniem narzędzia CASE ObjectTeam firmy Cadre. **(k)**

Oracle Workgroup/2000

Seminarium poświęcone zestawowi baz danych i narzędzi Workgroup/2000 firmy Oracle odbyło się w styczniu w Warszawie. Referowali przedstawiciele firm Oracle, Novell i Compaq. Oprócz samego produktu, omówiono zagadnienia będące przedmiotem współpracy partnerskiej wymienionych firm: m.in platformę sprzętową oraz miejsce baz danych Oracle Workgroup/2000 w globalnej koncepcji całościowych rozwiązań informatyzacyjnych firm i przedsiębiorstw. W programie seminarium znalazły się również demonstracje. Przybyło trzystu pięćdziesięciu zaproszonych gości.

Oracle Workgroup/2000 wychodzi naprzeciw zapotrzebowaniu drobniejszych firm, oferując rozwiązania dla zespołów od kilku do kilkudziesięciu stanowisk. Jego trzon stanowią jednostanowiskowa baza danych Personal Oracle 7 i wielodostępna baza danych Oracle 7 Workgroup Server. W skład oferty wchodzą również Oracle Objects for OLE, czyli narzędzia pozwalające na pełne wykorzystanie dostępu do baz danych przez aplikacje pisane w Borland C++, MS Visual Basic, Visual Basic for Applications i OLE 2.0, oraz Oracle Power Objects - pakiet narzędzi do graficznego, zorientowanego obiektowo tworzenia aplikacji dostępu do baz danych. Zestaw uzupełniają Oracle Mobile Agents, element pośredniczący pomiędzy serwerem i klientem, pozwalający na optymalne korzystanie z baz danych w przypadku podatnych na przerwy i zakłócenia połączeń z zastosowaniem telefonii kablowej lub komórkowej. Cały zestaw Workgroup/2000 ma cechować jakość rozwiązań, skalowalność, otwartość, łatwość użytkowania, całkowita zgodność z innymi produktami Oracle oraz przystępne ceny.

Wersję demonstracyjną oprogramowania Workgroup/2000 można zdobyć na serwerze WWW **http://www.oracle.com.** Z decyzją zakupu warto się jednak pospieszyć, do końca lutego produkty Oracle można kupić o 30% taniej.

(pb)

Uwaga członkowie PTI!

W związku z licznymi pytaniami wpływającymi do redakcji od członków PTI, uprzejmie informujemy, że od stycznia br. w ramach porozumienia zawartego między naszą redakcją a Zarządem Głównym PTI, wszyscy członkowie tego stowarzyszenia mają uprawnienia do prenumerowania INFORMATYKI w cenie ulgowej, tj. 3,75 zł za egzemplarz. Wystarczy na blankiecie wpłaty wpisać „członek PTI". Nie są wymagane dodatkowe potwierdzenia tego faktu przez władze PTI.

Laureaci konkursu na prace magisterskie

Do XII Ogólnopolskiego Konkursu Polskiego Towarzystwa Informatycznego na najlepsze prace magisterskie z informatyki zgłoszono w 1995 r. 11 prac wykonanych w roku akademickim 1994–1995 w ośmiu polskich wyższych uczelniach. Komisja konkursowa w składzie: prof. dr hab. Czesław Daniłowicz (przewodniczący), prof. dr hab. inż. Zbigniew Huzar, prof. dr hab. Jerzy Kucharczyk, prof. dr hab. Zygmunt Mazur, dr inż. Zbigniew Szpunar (sekretarz) – przyznała następujące nagrody:

– **I nagrodę** w wysokości 1000 zł otrzymali: **mgr inż. Andrzej Piotr Bodzek** i **mgr inż. Artur Paweł Chmielewski** za pracę pt. *„Przetwarzanie obrazów z wykorzystaniem technik drżenia kolorów"* (Politechnika Warszawska, Wydz. Elektroniki i Technik Informacyjnych, Instytut Informatyki, promotor: mgr inż. Grzegorz B. Mazur).

– **II nagrodę** w wysokości 800 zł otrzymali: **mgr inż. Cezary Jachniewicz** i **mgr inż. Paweł Wichniewicz** za pracę pt. *„Zastosowanie teorii zbiorów przybliżonych do analizy konfliktów"* (PW, Wydział Elektroniki i Technik Informacyjnych, Instytut Informatyki, promotor: prof. dr hab. Zdzisław Pawlak).

– **III nagrodę** w wysokości 600 zł otrzymał **mgr Roman Szczepan Kniter** za pracę pt. *„System PC-Expert - narzędzie do tworzenia prostych, dziedzinowo niezależnych systemów ekspertowych"* (Uniwersytet Gdański Wydz. Matematyki i Fizyki, Instytut Matematyki, promotor: dr Joanna Jędrzejowicz.

Autorom trzech nagrodzonych prac PTI ufundowało udział w Szkole PTI w Szczyrku w czerwcu 1996 r.

Przyznano ponadto **trzy równorzędne wyróżnienia** po 400 zł. Otrzymali je: **mgr inż. Maria Kasprzak** za pracę pt. *„Algorytm sekwencjonowania łańcuchów DNA"* (Politechnika Poznańska, Wydz. Elektryczny, Instytut Informatyki, promotor: prof. dr hab. inż. Jacek Błażewicz), **mgr Iwona Maksymowicz** za pracę pt. *„Implementacja w języku C zaproponowanego przez El-Gamala kryptosystemu z publicznym kluczem"* (UJ, Wydz. Matematyki i Fizyki, Instytut Informatyki, promotor dr hab. Wit Foryś), **mgr Janusz Woźniak** i **mgr Marcin Żelawski** za pracę pt. *„Rozproszony system pakowania figur 2-wymiarowych"* (UJ, Wydz. Matematyki i Fizyki, Instytut Informatyki, promotor: dr Maciej Ślusarek).

Gratulujemy Laureatom i zapraszamy do współpracy z naszą redakcją. **(kar)**

Informix – partnerstwo dla sukcesu

Dzień przed otwarciem targów Komputer Expo-96 odbyła się konferencja, na której ogłoszono nominację Piotra Mondalskiego na stanowisko dyrektora generalnego polskiego oddziału firmy Informix. Dotychczas istniało jedynie polskie biuro Informix Software Sp. z o.o., a jej szef był w randze dyrektora ds. sprzedaży. Utworzenie polskiego oddziału firmy Informix oznacza zdecydowane zwiększenie zainteresowania firmy tym regionem Europy i uzyskanie dużo większej samodzielności polskiej jednostki.

Nowy dyrektor generalny był dotychczas wiceprezesem i dyrektorem generalnym firmy PROKOM, która jest dystrybutorem oprogramowania oraz jednym z najsilniejszych partnerów Informixa w Polsce. Mając doświadczenie w realizacji wielkich projektów informatycznych, będzie obecnie kładł nacisk na dobór i rozwijanie sieci autoryzowanych partnerów Informixa w Polsce.

Rok 1995 był najlepszym w działalności Informixa w Polsce, kiedy to firma zanotowała ponad 100% wzrost obrotów. Do końca ubiegłego roku polskie biuro Informixa nawiązało współpracę i zawarło umowy ze 100 autoryzowanymi partnerami na terenie całego kraju. (J.S)

Nowe sterowniki firmy Xionics

Firma Xionics poleca dwa nowe produkty: podsystemy karty graficznej i monitora XipView Grayscale 1600i i XipView Color 1600i. XipView Grayscale jest przeznaczony do pracy w skali szarości z rozdzielczością 1600x1280 i częstotliwością odświeżania 76 Hz. Kompresja, dekompresja oraz rotacja są dokonywane sprzętowo, a nie przez oprogramowanie, co zwiększa dwu-, trzykrotnie szybkość pracy w porównaniu ze sterownikami stosowanymi tradycyjnie. Model XipView Color pracuje przy rozdzielczości 1600 × 1280 w 256 kolorach z palety 16,7 mln oraz z maksymalną częstotliwością odświeżania 85 Hz. Oba podsystemy są kompatybilne z oprogramowaniem współpracującym z ImageAccel firmy Cornerstone ze sterownikami ISIS, jak również z aplikacjami Windows. Dystrybutorem tych produktów jest warszawska firma ATTI Polska. **(k)**

Przypominamy Czytelnikom, że regularne otrzymywanie INFORMATYKI gwarantuje tylko jej prenumerata. Nasz miesięcznik można też kupić w kilku warszawskich księgarniach. Oto ich adresy i telefony:
ul. Mokotowska 51/53, tel.: 628-16-14;
ul. Świętokrzyska 14, tel.: 26-63-38 i 26-21-59;
ul. Targowa 61, tel.: 619-77-90;
ul. Mazowiecka 12, tel: 26-80-17 – Klub Prasy Technicznej.
Sprzedaż prowadzi również Dział Kolportażu Wydawnictwa SIGMA (sprzedaż przelewowa lub za zaliczeniem pocztowym), wystarczy tam zadzwonić – tel. 40-37-31, lub napisać: 00-716 Warszawa, ul. Bartycka 20, paw. B.
Zaprenumerować INFORMATYKĘ można w dowolnej chwili na dowolny okres. Wystarczy wpłacić dla Zakładu Kolportażu Wydawnictwa SIGMA na konto: PBK S.A. III O/Warszawa nr 3700-1573-139-11 wielokrotność ceny jednego numeru, tj. 5,0 zł przemnożone przez liczbę miesięcy, w których chcecie Państwo otrzymywać nasz miesięcznik.
Uczniowie szkół zawodowych i studenci szkół wyższych mogą zaprenumerować INFORMATYKĘ w cenie ulgowej – 3,75 zł za egzemplarz, ale powinni na blankiecie zamówienia uzyskać stempel swojej szkoły lub uczelni.
Natomiast członkowie Polskiego Towarzystwa Informatycznego otrzymują prenumeratę ulgową po wpisaniu na blankiecie zamówienia „członek PTI", bez żadnych dodatkowych pieczęci.
Mamy nadzieję, że to wyjaśnienie ułatwi Państwu zaprenumerowanie INFORMATYKI. W razie jakichkolwiek wątpliwości prosimy do nas dzwonić: tel. 619-11-61.

Redakcja

Integracja i partnerstwo

dokończenie z I s. okładki

powiedni sposób środowisko, w którym działa przedsiębiorstwo lub organizacja gospodarcza.

Targi Komputer Expo-96 potwierdziły tę diagnozę rozwijaną już od co najmniej dwóch lat. Coraz mniej mamy do czynienia z prostym oferowaniem pudełek i pudeł. Właściwie już prawie wszystkie firmy dodają do oferowanych systemów komputerowych wartość w postaci różnego typu oprogramowania własnego lub często niezwykle zaawansowanych pakietów specjalistycznych pochodzenia zagranicznego. W tym roku dało się zauważyć jeszcze więcej pakietów, w tym wiele polskich służących do zarządzania przedsiębiorstwem. Nareszcie polscy producenci oprogramowania zauważyli, że mogą na polu aplikacji bankowych, finansowych i wspomagających kierowanie firmą konkurować z oprogramowaniem zachodnim. Powód jest prosty - polskie pakiety są od razu dostosowane do obowiązujących tutaj regulacji prawnych. Z drugiej strony, coraz mniej firm zachodnich oferuje aplikacje, które nie są lokalizowane. To wszystko cieszy!

Najbardziej jednak radował mnie fakt, że wiele firm, przede wszystkim dużych zachodnich korporacji, otaczał wianek partnerów. Oznacza to konsolidowanie się wspólnych interesów wokół sprawy rozwijania dużych systemów informatycznych. Firmy przestają się wreszcie zachowywać jak pies ogrodnika - same nie poradzą, ale innemu też nie dadzą. Oznacza to także przekształcanie się polskiego rynku informatycznego. Kończą się czasy setek pecetów upychanych, gdzieś po kątach. Teraz firmy, które do tej pory pełniły rolę generalnych dostawców, przeradzają się w rasowych integratorów systemów. Chcąc wygrywać duże przetargi muszą nie tylko dostarczać pudełka, ale również tworzyć projekt systemu, budować sieć strukturalną, łączyć się z odległymi miejscami w Polsce, a nawet za granicą i wreszcie, co najważniejsze, wdrażać aplikację, która będzie spełniała wymagania użytkownika. Do realizacji tych wszystkich działań, w zakresie systemów komputerowych, które osiągnęły już w naszym kraju pewną wielkość krytyczną, niezbędni są partnerzy. To niezwykle ważny proces, ponieważ współpraca przy realizacji dużych systemów informatycznych daje polskim partnerom dużą wiedzę i doświadczenie. Na minionych targach ten proces było widać.

Co więcej, widać było, że duże firmy zaczynają ze sobą konkurować o partnerów. To też prawidłowy przejaw gospodarki wolnorynkowej. Trzeba zaś mieć świadomość, że dla coraz większej liczby firm zachodnich, które działają wyłącznie za pośrednictwem partnerów, ich stabilna współpraca z firmą macierzystą jest warunkiem odnoszenia sukcesów obu zainteresowanych stron.

Dzięki staraniom Centrum Promocji Informatyki z każdym rokiem rozwija się coraz bardziej idea cykli seminaryjnych towarzyszących targom. Niosą one ze sobą mnóstwo wiedzy nie tylko teoretycznej, ale przede wszystkim praktycznej. W większości przypadków seminaria były poświęcone omówieniu systemów informatycznych już wdrożonych i działających na potrzeby administracji państwowej, przemysłu, handlu, banków, towarzystw ubezpieczeniowych, wojska i policji. Do seminariów tym razem włączyły się aktywnie największe firmy bazo-danowe, jak chociażby Informix czy Oracle i największy producent oprogramowania - Microsoft. To tylko potwierdza ewolucyjny rozwój rynku informatycznego w kierunku dużych systemów. Redakcja „Informatyki" może tylko przyklasnąć zmianom idącym w tym kierunku, ponieważ ich tropienie mieści się w nowym profilu naszego pisma. O tym jednak napiszemy następnym razem.

Czy zatem Targi Komputer Expo-96 były udane? Zdecydowanie tak! Organizatorom trzeba pogratulować uporu, z jakim dążyli już po raz jedenasty do przeprowadzenia tej największej imprezy informatycznej w Polsce. Gratulować trzeba tym bardziej, że kierownictwo Biura Reklamy nie załamuje się różnymi pogłoskami, które każdego roku pojawiają się na temat rychłego końca tej formy promocji informatyki. Trzymać tak dalej!

Jerzy Szyller

Zaprosili nas...

Daniel Le Coguic, Dyrektor Generalny Bull Polska Sp. z o.o. na podsumowanie roku pracy firm Bull i Zenith Data System.

IBM Polska Sp.z o.o. na ostatnie spotkanie Antoniego Rozwadowskiego z przedstawicielami mediów, podsumowujące rok 1995 oraz poznanie nowego dyrektora generalnego IBM Polska – Mirosława Szturmowicza.

Oficyna Informatyczna i Fundacja Kultury Informatycznej na kolejne czwartkowe spotkanie na temat: „Rozwój zastosowań informatyki a kierunki szkoleń informatycznych".

KBN na pierwsze posiedzenie Zespołu Ekspertów, którego zadaniem jest wykonanie ekspertyzy dotyczącej sposobu konstruowania cenników za korzystanie z Internetu, wskazanie źródeł jego finansowania i kompleksowego rozwoju w Polsce.

Tomasz Czechowicz, prezes JTT Computer S.A. na lunch prasowy, podczas którego przedstawiono wyniki finansowe firmy w 1995 roku oraz zaprezentowano jej nową strategię i plany na rok bieżący.

Waldemar Sielski, dyrektor Microsoft Polska Sp. z o.o. na premierę polskiej wersji pakietu zintegrowanego „Microsoft Works dla Windows 95".

PROKOM na noworoczną konferencję prasową. W jej trakcie poinformowano o pozycji firmy na rynku, jej osiągnięciach oraz planach na przyszłość.

Firmy ATM i Hector na uroczystość otwarcia INTERNET CAFE w siedzibie firmy Hector – najpiękniejszym budynku Warszawy.

Centrum Prasowe Polskiej Agencji Informacyjnej SA w imieniu Biura Reklamy SA na konferencję prasową z okazji największej i najbardziej prestiżowej imprezy – XI Międzynarodowych Targów Komputer Expo '96.

Oracle na konferencję prasową i seminarium poświęcone nowej serii produktów „Oracle Workgroup 2000".

Techmex SA na tradycyjne, doroczne spotkanie partnerów, w trakcie którego przedstawiono strategię handlową firmy w roku 1996.

SOLIDEX w imieniu swojego partnera Sun Microsystems na seminarium pt. „To nie sen, to Java", poświęcone prezentacji rewolucyjnego języka Java, który odmieni Internet.

BPS na konferencję prasową promującą nowe produkty – zmiennoprądowy zasilacz awaryjny PowerWorks A40 oraz system transmisji danych HDSL WATSON.

IBM Polska na spotkanie pod nazwą „Parapetówa – czyli IBM od kuchni" w przededniu otwarcia Targów Komputer Expo '96.

Computer Integration Technology Publishing Institute na dni otwarte swojego biura oraz cykl seminariów prowadzonych przez specjalistów firm Kodak, Linotype-Hell, PAP oraz IDG.

ComputerLand Poland S.A. i Przedsiębiorstwo Techniki Bankowej elba Spółka z o.o. na konferencję prasową informującą o wspólnym przedsięwzięciu stwarzającym nowe możliwości obsługi sektora bankowego. **(t)**

Voxel Analyst – pakiet dla badacza

Firma Intergraph Corp. wprowadziła na rynek pakiet Voxel Analyst pracujący w systemach Windows 95 i Windows NT. Pakiet umożliwia faktyczne modelowanie, analizę i wizualizację masowych, trójwymiarowych rozproszonych danych naukowych. Jest to pierwszy produkt dla komputerów klasy PC, który umożliwia analizę boolowską i interaktywne przetwarzanie danych masowych. Wykorzystuje on platformę graficzną OpenGL. Wielowątkowość umożliwia równoczesne wykonywanie obliczeń i wyświetlanie wyników na maszynach wieloprocesorowych z szybkością dostępną dotąd jedynie na najlepszych graficznych stacjach roboczych. Pakiet może importować dane w formatach wielu popularnych pakietów CAD, GIS i aplikacji biurowych oraz ASCII. Operacje analityczne, takie jak przekroje czynne, płaszczyznowe, przejścia siodłowe, izopowierzchnie i izobary mogą być interaktywnie generowane i zmieniane. (k)

AVK 5 i Amasys 1.0 – po polsku

Znany w Europie Zachodniej niemiecki program antywirusowy AVK 5 (AntiVirenKit 5) wprowadza na nasz rynek w polskiej wersji językowej firma ze Szczecinka – LTP MediaSoftware-G Data Software, która jest przedstawicielem niemieckich firm o tej samej nazwie. AVK 5 został wyróżniony na targach SOFTARG '95 w Katowicach. Wykrywa on i usuwa ponad 8 tys. wirusów, łącznie z wirusami polimorficznymi i ich mutacjami. Innym programem niemieckim, który proponuje firma ze Szczecinka, jest Amasys 1.0 – uniwersalny, interaktywny program zarządzający bazami danych (adresy, materiały itp.) wraz ze wszystkimi numerami kodów pocztowych, numerami kierunkowymi oraz numerami banków – dla Polski i Niemiec, a także innych krajów – po dołączeniu odpowiedniej bazy. Obydwa programy pracują pod Windows 3.1 i Windows 95. W marcu na targach CeBIT 96 zostanie przedstawiona 32-bitowa wersja programu AVK6. (k)

Netscape dla systemu Solaris

Firma Sun ogłosiła wprowadzenie na rynek rodziny przeglądarek i serwerów Netscape przeznaczonych dla komputerów zarządzanych przez system operacyjny Solaris. W skład rodziny wchodzą: Netscape Communications Server v. 1.12, Netscape News Server v. 1.12, Netscape Navigator v. 1.12 i Netscape Proxy Server v 1.12. Oprogramowanie Netscape może teraz bezproblemowo współpracować z innym oprogramowaniem oferowanym dla platformy Solaris, włączając w to oprogramowanie zapewniające mechanizmy zabezpieczające z grupy produktów Solstice, produkty przeznaczone do integracji środowisk operacyjnych PC/Internet SolarNet i oprogramowanie serwerów Netra. (k)

Nowe Vectry HP

Hewlett-Packard Vectra XU to nowa seria komputerów przeznaczona przede wszystkim dla zastosowań w tych dziedzinach, które wymagają dużych mocy obliczeniowych, takich jak inżynieria oprogramowania, DTP, CAD. HP Vectra XU 6/150 model 2000 jest wyposażony w dwa procesory Pentium Pro pracujące z częstotliwością 150 MHz, dysk twardy 2 GB, 16 MB pamięci RAM z możliwością powiększenia do 256 MB. Karta grafiki – Matrox Millenium – zawiera standardowo 2 MB pamięci obrazu i może być rozszerzona do 4 MB, co zapewnia maksymalną rozdzielczość obrazu 1600 × 1200 punktów przy palecie 64 tys. barw. Komputer ma sześć gniazd rozszerzeń. (k)

Osiągnięcia CSBI

Jak wynika ze wstępnych szacunków, obroty holdingu Computer Systems For Business International (CSBI SA w Polsce i CSBI EE w Rosji) wyniosły w 1995 r. 33 mln zł i były wyższe o ok. 67% od obrotów z 1994 r. Około 66% obrotów stanowiła sprzedaż profesjonalnego oprogramowania i usług związanych z jego wdrożeniem oraz języka czwartej generacji i oprogramowania do tworzenia relacyjnych baz danych Progress 4GL RDBMS. Obecnie w ponad 200 polskich bankach funkcjonuje system BANKIER, jest on też wdrożony w jednym z największych rosyjskich banków TOKO BANK. Sukcesy odnoszą też systemy zarządzania CSBI – PRO/MIS i MPR II.

Turbolasery z Digitala

Turbolasery to najnowsza rodzina dużych serwerów serii 8 × 00 Digitala. Wyposażone są one w 64-bitowe procesory Alpha. Każdy Turbolaser może zawierać 6 do 12 procesorów w wersjach od 100 do 300 MHz. Serwery te mogą obsługiwać banki pamięci masowych o wielkościach do 1 TB. Pozwala to na wczytywanie dużych baz danych i dokonywanie operacji bezpośrednio w pamięci operacyjnej, co znacznie przyspiesza szybkość przetwarzania informacji. Prezentowany na targach Komputer Expo-96, w stoisku Digitala Turbolaser należał do klienta Digitala – Telekomunikacji Polskiej SA. Inny Turbolaser – AlphaServer 8200 z systemem OpenVMS – został zakupiony dla Biura Projektowego Stoczni Gdyńskiej. (k)

Nagrody Wydawnictwa LUPUS

Już po raz szósty Wydawnictwo LUPUS wręczyło polskim produktom swoje nagrody (oprogramowanie i sprzęt). Nagrody dwutygodnika PCkurier – tytuły **„Produkt Roku '95"** – otrzymały: **EuroPlus+** – multimedialny, interaktywny kurs języka angielskiego, gdańskiej firmy Young Digital Poland; **Fryderyk Chopin** – multimedialna encyklopedia wiedzy o życiu i twórczości Fryderyka Chopina, krakowskiej firmy NeuroSoft; **QR-Tekst 2.0 dla Windows** – edytor tekstów warszawskiej firmy Malkom oraz rodzina **UPS-ów Fideltronic PRO**, produkowana przez firmę Fideltronik z Zembrzyc. Przyznaną po raz pierwszy Nagrodę **„Wybór Czytelników"** PCkuriera otrzymały: system **Windows 95 PL** firmy Microsoft i komputer **Optimus Workstation 6P150** z procesorem PentiumPro. Tytuły **Produkt Roku '95** Magazynu AMIGA czytelnicy przyznali: w kategorii sprzętu przystawce **ProModule** warszawskiej firmy Elsat s.c, a w kategorii oprogramowania programowi **AMiTextPro** krakowskiej firmy Twin Spark Soft. W wewnętrznym konkursie Wydawnictwa na najlepszy artykuł opublikowany na łamach czasopism LUPUSA, tytuł **„Pióro Roku '95"** otrzymał **Jan Stożek** za artykuł „Bądź mądry przed szkodą" opublikowany w „Enterze" 10/95. (k)

Polskie firmy na CeBIT '96

W tegorocznych targach CeBIT w Hanowerze (14–20 marca) weźmie udział 16 polskich firm. Już po raz drugi jedzie do Hanoweru sześć firm: **COMPUTER Systems for Business International** ze Zintegrowanym Systemem Bankowym, **DGT** – producent urządzeń i sprzętu telekomunikacyjnego, firma **Inwar** z Sieradza – producent komputerów i notebooków, **Super-Memo World** – miedzy innymi ze swoim najnowszym produktem Genius, **Young Digital Polska** – producent multimedialnych produktów do nauki języków obcych oraz **2SI Sieciowe Systemy Informacyjne** (d. Techmex International) – integrator systemów informatycznych. Zadebiutują w tym roku: **Unicorn, Elzab, Arkus Electronics, Telbank, Warnet, Softbank, Suntech, Contec, Apexim i Qumak International.** Polska ekspozycja zajmie w sumie 177 m kw. powierzchni. Udział w tych targach jest bardzo kosztowny, wystawcom pomaga Departament Promocji Ministerstwa Współpracy Gospodarczej z Zagranicą, który dofinansowuje w 50% koszt wynajęcia powierzchni targowej. Organizacją zajmuje się Polska Izba Informatyki i Telekomunikacji oraz Centrum Promocji Informatyki. (k)

Wiosenna Szkoła PTI

Tegoroczna Wiosenna Szkoła PTI odbędzie się w dniach 13–17 maja w ośrodku „Dolna Odra" w Świnoujściu pod hasłem „Szybkie wytwarzanie aplikacji – Organizacja, metody, narzędzia".
Organizatorzy przyjmują zgłoszenia i sugestie związane z organizacją szkoły. Kontakt: Koło PTI w Szczecinie, 71-101 Szczecin, ul. Mickiewicza 64, prof. dr hab. Zdzisław Szyjewski, faks (091) 87 88 64, e-mail, zszyew@uoo.univ.szczecin.pl (k)

Gulipin Computers

Firma Gulipin Computers, z siedzibą we Wrocławiu, rok 1995 zakończyła obrotem w wysokości 11 mln zł netto, co oznacza wzrost o 40%, Gulipin wyprodukował 33 540 sztuk komputerów, rozpoczął też montaż i sprzedaż komputerów wieloprocesorowych. (k)

Technika Informatyczna motorem Twojego businessu

Żadna impreza na świecie nie dorównuje CeBIT '96 w Hanowerze pod względem kompletności i aktualności prezentacji Techniki Informatycznej. Niepowtarzalny pokaz światowego rynku Techniki Informatycznej skoncentrowany w 25 nowoczesnych i funkcjonalnych halach wystawienniczych. CeBIT'96 w Hanowerze, największe na świecie targi Techniki Informatycznej, to demonstracja śmiałych, innowacyjnych rozwiązań, najnowocześniejszych produktów i szerokiego wachlarza usług. To wyznacznik trendów rozwojowych dla całej branży. Ponad 6.000 wystawców z 60 krajów ze wszystkich kontynentów nie szczędzi wysiłków aby jak najatrakcyjniej przedstawić swoją ofertę i zainteresować profesjonalnych odbiorców. Wykorzystaj tę jedyną okazję!

Technika Informatyczna

Network Computing

CIM

Software

Telekomunikacja

Technika Biurowa

Technika Bankowa i Technika Obrotu Pieniężnego

Technika Zabezpieczeń

Badania

Fax +49 511 288 86 50, Internet: http://www.messe.de

DEUTSCHE MESSE AG, HANNOVER/GERMANY
Bliższe informacje:
Ewa Samkowska, Przedstawiciel w Polsce,
Deutsche Messe AG, Hannover, ul. Mokotowska 49, 00-542 Warszawa,
Tel.: (022) 66 00-505, 66 00-525, Teleks: 814241 mex pl,
Telefaks: (022) 29 85 59

Hannover 14.-20.3.1996

CeBIT

Światowe Centrum Techniki - Biurowej - Informatyki - Telekomunikacji

Firma RAD Data Communications wprowadza nowy modułowy koncentrator dostępu do ATM – ACE-2000, zaprojektowany dla potrzeb operatorów telekomunikacyjnych i sieci w większych przedsiębiorstwach. Dzięki zaimplementowaniu warstw adaptacyjnych AALI (ATM Adaptation Layers), AAL3/4 i AAL5, ACE-2000 pozwala użytkownikom na korzystanie z szerokiej gamy usług CBR (stała przepustowość), VBR (zmienna przepustowość) i ABR (dostępna przepustowość).

*

Koncern Bull, który do września ub. r. należał do francuskiego Skarbu Państwa, po raz pierwszy od lat osiągnął zysk (119 mln USD). W ub. r. Bull został sprywatyzowany; obecnie tylko 36,5% udziałów ma Skarb Państwa, pozostałe należą do France Telekom, Motoroli i NEC-a. Bull, który w Polsce jest znany głównie z budowy systemu Poltax, podpisał z Ministerstwem Finansów – jak poinformował szef polskiego oddziału firmy p. Daniel Le Coguic – nowy kontrakt. Opiewa on na kwotę 2,5 mln franków i dotyczy dostawy sprzętu.

*

Centrum Edukacyjne EduSoft Sp. z o.o. jako pierwsze w Polsce, w oparciu o polskie materiały szkoleniowe uruchamia szkolenia z zakresu systemu operacyjnego SCO Openserver 5.0. Adaptacji i tłumaczenia materiałów szkoleniowych do polskich warunków dokonali pracownicy Centrum EduSoft.

*

Firma JTT Computer S.A. odnotowała w 1995 r. obroty w wysokości 173 mln PLN, co stawia ją na drugim miejscu wśród polskich firm komputerowych. Zysk netto wyniósł ponad 5,5 mln PLN zł. Obroty całej grupy JTT (nieskonsolidowanej) wyniosły 325 mln PLN. Pod koniec 1995 r. JTT wydzieliła ze swojej struktury Pion Kas Fiskalnych w postaci osobnej spółki – JTT Silesia Sp. z o.o., której siedzibą jest Zabrze. Produkuje ona kasy fiskalne Entry 01 konstrukcji IBM.

*

Według danych International Data Corporation, renomowanej amerykańskiej firmy zajmującej się badaniami rynku, firma Hewlett-Packard zajęła pierwsze miejsce w dziedzinie programów do zarządzania sieciami. Udział programu HP OpenView osiągnął 35,3%, zajmująca dotychczas pierwsze miejsce firma Sun ma 32,7% udziału w rynku, trzecie miejsce zajmuje firma Cabletron (13%), czwarte IBM (8,9%). HP przystąpił do dystrybucji najnowszej wersji programu zarządzającego, przeznaczonego dla systemu klient-serwer – OpenView Network Node Manger 4.0.

*

Oracle Corporation ogłosiła, iż w II kwartale roku fiskalnego 1995 (kończącym się 30 listopada 1995 r.) obroty firmy wzrosły o 44%, do rekordowej wartości 967 mln USD. Zyski koncernu w tym okresie wyniosły 137 mln USD, co oznacza wzrost o 48% w stosunku do analogicznego okresu minionego roku.

*

NetWare firmy Novell, jak wykazują badania firmy Dataquest, jest najpopularniejszym w Europie sieciowym systemem operacyjnym. Oblicza się, że stanowi on 66% rynku. Drugie miejsce zajmuje IBM - 13% rynku. Dochody Novella ze sprzedaży NetWare 4.1 stanowią 50% sprzedaży wszystkich produktów NetWare.

*

Intel wprowadził do sprzedaży hurtowej procesory Pentium o częstotliwości taktowania 150 i 166 MHz. Procesor 166 MHz osiąga 4,76 SPECint95 i 3,37 SPECfp95 w operacjach zmiennopozycyjnych. Równocześnie z ogłoszeniem przez Intela wprowadzenia nowych procesorów firma Dell wprowadziła na rynek dwa nowe modele komputerów z serii OptiPlex, wyposażone w te procesory: OptiPlex GX5150 (Pentium 150 MHz) oraz OptiPlex GX5166 (Pentium 166 MHz).

*

Digital Equipment Corporation przedstawiła nowe oprogramowanie o nazwie FX!32, składające się z translatorów i emulatorów umożliwiających pracę oprogramowania dla Windows na komputerach AlphaGeneration. Czasopismo BYTE przyznało temu oprogramowaniu tytuł „Najlepsza technologia". Technologia FX!32 poddawana jest obecnie wielu testom, na rynku powinna pojawić się w połowie 1996 r.

*

Firma Sony rozpoczęła produkcję pierwszej kasety przeznaczonej do gromadzenia danych typu Metal QIC-WIDE. Oznaczona symbolem QWX 3210XL kaseta ma pojemność rzędu 2.3 GB (4.6 GB po kompresji). Nowe rozwiązanie zostało oparte na formacie QIC 163, zaakceptowanym przez Komitet Standaryzacyjny QIC (Quarter-Inch-Cartridge – kaseta zawierająca taśmę magnetyczną o szerokości 0,25 cala). Firma Tandberg Data wprowadziła już do sprzedaży pierwszy napęd pozwalający na wykorzystanie nowej technologii Metal QIC-WIDE: Panther Mini 4600.

*

Microsoft wprowadził obsługę głosu do programu obsługi Unimodem w Windows 95. Programiści mogą uzyskać nowy program obsługi Unimodem V dostępny bezpłatnie za pośrednictwem Internetu i najważniejszych serwisów sieciowych. Ulepszony program pozwala na łatwiejsze i szybsze tworzenie programów komunikacyjnych dla Windows 95, korzystających zarówno z modemów głosowych, jak i wielu usług interfejsu programisty Telephony API (TAPI).

*

Obroty gdyńskiej firmy PROKOM zajmującej szóste miejsce w rankingu największych polskich firm komputerowych Top 200, wzrosły z 63,80 mln PLN w 1994 r. do 117,89 mln PLN w 1995 r. PROKOM był pierwszym polskim dystrybutorem oprogramowania Informixa, opracował jego pierwszą wersję. Obecnie największym kontraktem PROKOMU jest komputeryzacja PZU SA; rozpoczęta we wrześniu 1995, ma być zakończona w 1998 r.

*

Firma RAD Data Comunications wprowadziła na rynek ASM-MN-214, nowe dziewiętnastocalowe gniazdo modemowe, do którego można podłączyć do czternastu dwużyłowych, czterożyłowych i światłowodowych kart modemowych. Karty te są modularnymi wersjami modemów krótkiego zasięgu ASM i modemów światłowodowych FOM firmy RAD.

*

Fińska firma Fiskars – znany producent UPS-ów wprowadziła na rynek zestaw składający się zasilacza awaryjnego do Windows 95 i oprogramowania do zarządzania poborem energii. Zestaw ten, zawierający nową wersję programu Fail-Safe III oraz zasilacz PowerRite Plus 400 VA lub 600 VA jest w pełni zgodny z Windows 95. Jest dostępny także w wersjach dla Windows NT, Windows 3.x i IBM OS/2. Dystrybutorem Fiskarsa w Polsce jest firma BPS.

*

Firma ICL oferuje nowy serwer z rodziny Ci, zaprojektowany jako rodzina serwerów plików i wydruków sieci LAN. Jest on wyposażony w Pentium z procesorem 100 MHz, dysk twardy 2 GB, 32 MB RAM. Zainstalowano go na Wydziale Matematyki i Informatyki Uniwersytetu Poznańskiego.

*

Borland International wypuściła na rynek najnowszą wersję popularnego pakietu do zarządzania bazami danych – Paradox 7 dla Windows 95 i Windows NT. Zapewnia ona obecnym użytkownikom Paradoxa bezbolesne przejście do środowiska Windows 95.

*

3Com przedstawiła nowe produkty ISDN do kompletnych systemów końcowych (end-to-end): Aperature II – urządzenie umożliwiające zdalne połączenia, 3Com AccessBuilder 4000 ISDN zdalnego dostępu i 3Com Arpeggio most/router. Współpracują one z dostępnymi już urządzeniami 3Com.

*

Seria narzędzi firmy Intergraph do obsługi grafiki rastrowej I/RAS została wzbogacona o program Power Ras, służący do konwersji grafiki rastrowej na wektorową. Power Ras oferuje nowe narzędzia edycji i konwersji, które współpracują m.in. z narzędziami wektorowymi MicroStation 95. Umożliwia także rastrowe wyświetlanie plików, edycję i konwersję do grafiki wektorowej. Pracuje w środowiskach Windows 95 i Windows NT.

*

Krakowska firma ASCOMP S.A. podpisała umowę z firmą Allied Telesyn Int. i dzięki temu uzyskała miano integratora sprzętu sieciowego tej znanej amerykańskiej firmy. Oferta firmy ASCOMP jest skierowana na rynek małych i średnich sieci komputerowych.

*

Techmex International Sp. z o.o. z dniem 1 lutego br. zmieniła swoją nazwę na Sieciowe Systemy Informacyjne Sp. z o.o. Firma, która przyjęła logo 2SI, od ponad trzech lat zajmuje się integracją systemów informatycznych. 2SI ma poza centralą w Katowicach pięć oddziałów. Obroty 2SI w 1995 r. przekroczyły 40 mln zł. Firma zatrudnia blisko 100 pracowników, prawie wyłącznie inżynierów.

Opracowała KRYSTYNA KARWICKA-WYCHLEWICZ

publikacje

ARIS-Toolset - narzędzie dla Business Process Reengineering

Mirosława Lasek
Katedra Cybernetyki i Badań Operacyjnych
Wydział Nauk Ekonomicznych
Uniwersytet Warszawski

ARTYKUŁ REKOMENDOWANY

Koncepcja *Business Process Reengineering* (BPR), oznaczająca potrzebę radykalnego przekonstruowania procesów ekonomicznych i struktur organizacyjnych przedsiębiorstw zdobywa coraz większą popularność. Lawinowo narasta liczba publikacji na ten temat [3, 6, 12]. Jednakże ze względu na „rewolucyjny" charakter zmian, jakie ze sobą niesie oraz złożoność procesów i struktur, których dotyczy, jej realizacja w praktyce nie jest łatwa. Pomocne okazuje się zastosowanie odpowiedniego oprogramowania wspomagającego proces *Business Reengineering*. Obecnie przeznaczonych do tego celu programów, zwanych programami BPR, jest już wiele. Według danych przedstawianych przez GartnerGroup, w tegorocznym raporcie, są one oferowane przez ponad 60 dostawców, a dochody na rynku programów BPR szacuje się na 100 mln dolarów [5].

Jednym z programów BPR, wymienianym w raporcie GartnerGroup, jest program **ARIS-Toolset** niemieckiej firmy **IDS Prof. Scheer GmbH**. Jest to program, który zdobył wielu użytkowników w krajach Europy Zachodniej i USA. Liczba sprzedanych licencji osiągnęła w sierpniu tego roku powyżej trzech tysięcy [7].

Użytkownikami ARIS-Toolset jest m.in. ponad 60 wyższych uczelni (w 15 krajach). Wydział Nauk Ekonomicznych Uniwersytetu Warszawskiego jest użytkownikiem programu od ponad roku. Program został zakupiony w ramach realizacji projektu Tempus: *„Kooperacja uniwersytecka – informatyka gospodarcza"*, finansowanego przez Europejską Wspólnotę Gospodarczą. Celem projektu było uruchomienie studiów z zakresu informatyki gospodarczej na Uniwersytecie Warszawskim. Program ARIS-Toolset jest wykorzystywany zarówno dla celów dydaktycznych, jak i potrzeb badawczych. Ze względu na jego oryginalność, szeroki i stale rosnący krąg użytkowników, do których należą przedsiębiorstwa przemysłowe, uniwersytety, firmy konsultingowe, producenci oprogramowania, a także fakt, że program ten nie jest szerzej znany w Polsce, chciałabym przedstawić go czytelnikom „Informatyki"[1].

Rekomenduje:
Prof. dr hab. Tadeusz Kasprzak
Kierownik Katedry Cybernetyki i Badań Operacyjnych
Wydziału Nauk Ekonomicznych
Uniwersytetu Warszawskiego

Zasady modelowania

Za pomocą ARIS-Toolset można opisywać i analizować istniejące procesy ekonomiczne i struktury organizacyjne (opis rzeczywistości), jak i projektować nowe procesy i struktury.

Zasady, jakie proponuje się w ARIS-Toolset do opisu i analizy istniejących procesów i struktur oraz projektowania procesów i struktur, są jednakowe. W obu przypadkach tworzone są modele przedstawiające problem z czterech punktów widzenia: danych, funkcji, organizacji

[1] W numerze 9 „Informatyki" z 1994 r., w artykule „ARIS – Architektura Zintegrowanych Systemów Informacyjnych", którego autorami są A.-W. Scheer, T. Elsner, D. Dziuba, M. Lasek [11], przedstawiono podejście metodyczne ARIS (niem. *Architektur Integrierter Informationssysteme*, ang. *Architecture of Integrated Information Systems*), opracowane przez prof. A.-W. Scheera z Uniwersytetu Saarlandzkiego w Saarbrücken [9, 10], zastosowane w programie ARIS-Toolet. Jednakże w artykule tym uwagę skupiono na samej metodyce ARIS. Ze względu na ograniczone ramy artykułu i brak jeszcze wówczas większego doświadczenia w użytkowaniu ARIS-Toolset przez polskich współautorów, samemu programowi poświęcono niewiele miejsca, zaledwie wspominając o jego istnieniu.

oraz sterowania (przedstawiającego powiązania między danymi, funkcjami, organizacją), a w ramach każdego z nich na trzech różnych poziomach abstrakcji: definiowania wymagań użytkownika (ang. *Business Concept*), specyfikacji projektowej (ang. *Data Processing Concept*), technicznej implementacji (ang. *Implementation*) [8, 9, 10].

Modele tworzone z różnych punktów widzenia i na różnych poziomach abstrakcji nie pozostają w izolacji, lecz są ze sobą powiązane, w ten sposób, że użytkownik może swobodnie przechodzić od modelu do modelu.

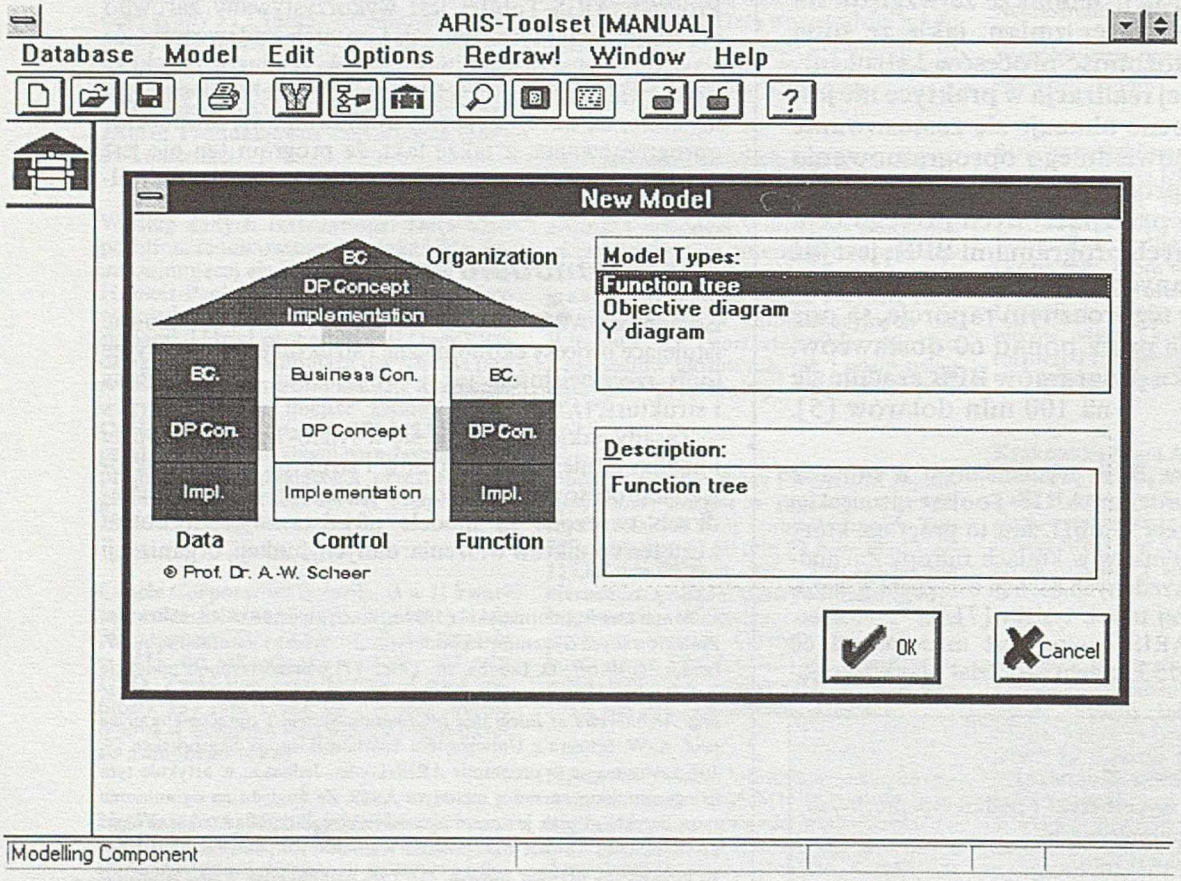
Poruszanie się wśród modeli, przedstawiających problem z różnych punktów widzenia i na różnych poziomach abstrakcji, ułatwia towarzyszący uytkownikowi programu „domek ARIS", tak jak zilustrowano to na rysunku 1, przedtawiającym ekran po wyborze polecenia dla utworzenia nowego modelu. Na przedstawionym „domku ARIS" wystarczy wskazać (kliknięcie myszą) jakiego punktu widzenia i jakiego poziomu ma dotyczyć model. W przedstawionym na rysunku 1 wypadku tworzenia modelu wybrano punkt widzenia funkcji (ang. *Function*) i najwyższy poziom: definiowanie wymagań użytkownika BC (ang. *Business Concept*). Ten fragment „domku ARIS" jest podświetlony. Program wyświetla typy modeli jakie możemy utworzyć dla wybranego punktu widzenia i poziomu. W rozpatrywanym przypadku są to: drzewo funkcji (ang. *function tree*), diagram celów (ang. *objective diagram*), diagram Y (ang. *Y diagram*). Wyświetlany jest też krótki opis aktualnie wybranego (podświetlonego) typu modelu. Po wybraniu typu modelu i przycisku OK (lub Enter) otwiera się ekran, umożliwiający utworzenie modelu żądanego typu.

W przypadku tworzenia drzewa funkcji jest dostępny tylko jeden typ obiektu: funkcja, przedstawiana graficznie za pomocą prostokąta o zaokrąglonych rogach i ikona umożliwiająca tworzenie powiązań między funkcjami. Aby wprowadzić obiekt wystarczy kliknąć ikonę obiektu, przemieścić wskaźnik myszy w miejsce obszaru, gdzie ma być umieszczony obiekt i kliknąć lewym przyciskiem myszy. Wprowadzenie powiązań jest równie proste.

Po nadaniu obiektom atrybutów, co następuje przez wskazanie dla każdego atrybutu jego typu i podanie wartości, definicje obiektów są zapisywane w bazie danych. Atrybuty możemy nadawać także relacjom. Możliwe jest dziedziczenie wartości atrybutów przez obiekty, np. wybranego typu w wybranych modelach, wybranego typu w tworzonym (bieżącym) modelu.

W ARIS-Toolset pod pojęciem bazy danych rozumie się składnicę informacji (ang. *repository*). Zawiera ona informacje o używanych metodach (meta-dane i meta-struktury), a także modele opracowane przez użytkownika (dane użytkownika). Opracowywane przez użytkownika definicje obiektów i modele są gromadzone, w bazie w grupach (ang. *groups*), których zasady tworzenia są analogiczne do zasad tworzenia katalogów w DOS.

Dzięki przechowywaniu definicji każdego obiektu w bazie, staje się możliwe zachowywanie kompletności i spójności opisów w każdym momencie modelowania. Obiekt, którego definicja znajduje się w bazie, może występować w różnych modelach, np. funkcje z drzewa funkcji mogą występować w modelach procesów. Zmiana atrybutów takiego obiektu w dowolnym modelu, zmienia jego definicję w bazie, a w ślad za tym dla każdego jego wystąpienia w dowolnych, innych modelach.



Rys. 1.
Ekran z „domkiem ARIS" po wyborze polecenia utworzenia nowego zbioru

Graficzne formy prezentacji modeli (diagramy)

Modele są budowane i przedstawiane w postaci diagramów, na których obiekty i relacje między obiektami są reprezentowane za pomocą symboli graficznych. Wykorzystywane są przykładowo diagramy: drzewa funkcji (ang. *function trees*), schematy organizacyjne (ang. *organigrams*), rozszerzone diagramy obiekt-relacja (ang. *expanded entity-relationship diagrams Chena – eERM*), rozszerzone łańcuchy procesów sterowane zdarzeniami (ang. *extended event-controlled process chains – eEPC*), diagramy przepływu danych (ang. *data flow diagrams*).

Na rysunku 2 przedstawiono przykład diagramu eERM. Został on sporządzony za pomocą ARIS-Toolset i jest dostarczany wraz z zakupionym oprogramowaniem w ramach opracowanego przez firmę IDS przykładu dydaktycznego. Dotyczy on realizacji zleceń w przedsiębiorstwie przemysłowym.

Posługiwanie się diagramami ułatwia możliwość jednoczesnego wyświetlania wielu diagramów na ekranie w oddzielnych oknach. Zasady poruszania się między oknami i obsługi okien są takie same jak we wszystkich programah działających pod Windows.

W programie mogą być tworzone hierarchie wzajemnie podporządkowanych sobie diagramów (modeli). Można przypisać diagram wybranemu obiektowi, np. funkcji (obiekt) może być przypisany diagram przedstawiający opisujące ją drzewo podfunkcji lub łańcuch procesów dla realizacji tej funkcji.

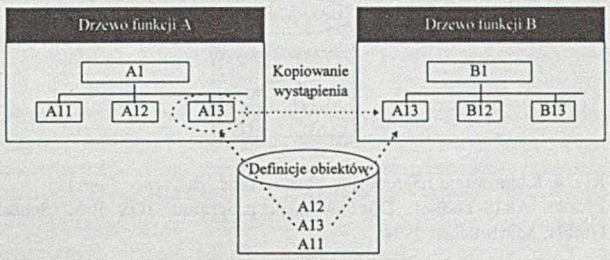
Kopiowanie obiektów

Obiekty mogą być kopiowane w ramach modelu lub do innego modelu tego samego typu (ten sam „punkt widzenia" i poziom) lub innego typu, jeśli tylko dopuszczone są w nim obiekty tego samego typu jak kopiowany.

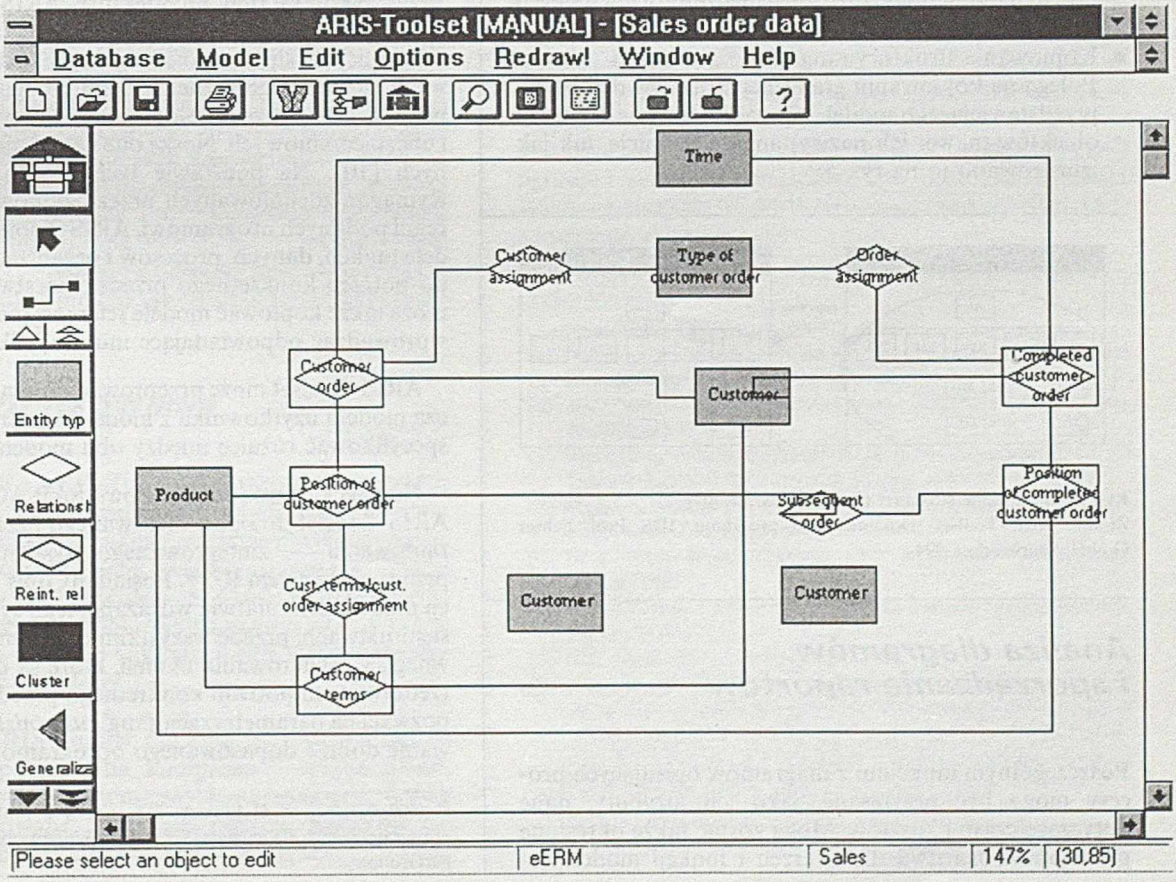
Dopuszcza się trzy rodzaje kopiowań obiektów:

- Kopiowanie wystąpienia obiektu (ang. *Occurence Copy*)

  Przy takim kopiowaniu nie jest tworzona nowa definicja dla skopiowanego obiektu. Istnieje tylko jedna definicja obiektu, który może występować w wielu miejscach modelu lub w wielu modelach, tak jak ilustruje to rysunek 3. Na rysunku przedstawiono przykład kopiowania obiektu A13 z diagramu modelu



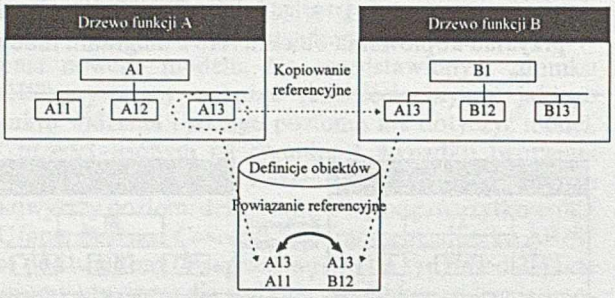
Rys. 3. Kopiowanie wystąpienia obiektu (ang. *Occurence copy*)
Źródło: ARIS-Toolset. Dokumentacja programu, IDS Prof. Scheer GmbH, Saarbrücken 1994



Rys. 2. Diagram eERM

drzewa funkcji *A* do diagramu modelu drzewa funkcji *B*. Nie została utworzona nowa definicja dla skopiowanego obiektu. Obiekt – oryginał i jego kopia mają jedną definicję w bazie.
Jeżeli zmienimy wartość atrybutu dowolnego wystąpienia obiektu, to zmieni się ona dla wszystkich obiektów, ponieważ mają jedną wspólną definicję. Dotyczy to także zmiany nazwy obiektu.

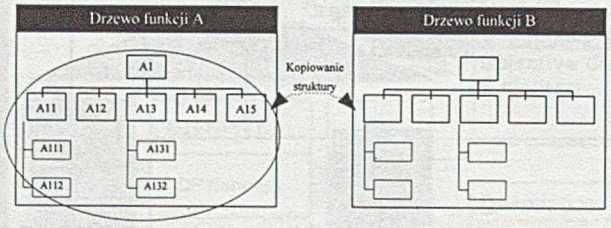
- Kopiowanie referencyjne (ang. *Reference Copy*)
W wypadku kopiowania referencyjnego dla kopii obiektu zostaje utworzona nowa definicja w bazie. Ilustruje to rysunek 4.



Rys. 4. Kopiowanie referencyjne obiektu (ang. *Reference Copy*)
Źródło: ARIS-Toolset. Dokumentacja programu, IDS Prof. Scheer GmbH, Saarbrücken 1994

Obiekty mają różne definicje. Istnieje powiązanie między tymi definicjami (ang. *Reference Link*), pozwalające zidentyfikować oryginał i kopię.
Jeżeli zmienimy wartość atrybutu skopiowanego obiektu, to nie zmieni się wartość tego atrybutu dla obiektu – oryginału.

- Kopiowanie struktury (ang. *Structure Copy*)
Polega na kopiowaniu graficznej struktury diagramu, przedstawiającego model. Nie są kopiowane atrybuty obiektów (nawet ich nazwy) ani ich definicje, tak jak zilustrowano to na rys. 5.



Rys. 5. Kopiowanie struktury (ang. *Structure Copy*)
Źródło: ARIS-Toolset. Dokumentacja programu, IDS Prof. Scheer GmbH, Saarbrücken 994

Analiza diagramów i sporządzanie raportów

Poszczególnym funkcjom z diagramów opisujących procesy mogą być przypisane, jako ich atrybuty, dane dotyczące czasu i kosztów. Mogą zostać także określone prawdopodobieństwa dla zdarzeń i funkcji modelowanych procesów. Na podstawie tych informacji o czasie, kosztach i prawdopodobieństwach ARIS-Toolset przeprowadza analizę dotyczącą czasu, kosztów i prawdopodobieństw dla całych procesów lub ich podprocesów, zapoczątkowywanych realizacją określonego zdarzenia (zdarzeń) i kończących się realizacją określonego zdarzenia (zdarzeń). Mogą być także analizowane inne wskaźniki ekonomiczne [2].

Możliwe jest generowanie raportów z bazy ARIS-Toolset. Raport może obejmować wybrane modele lub obiekty, lub wybrane grupy modeli i obiektów. Raport przedstawia w postaci tekstowej wyniki analizy diagramów (modeli), obiektów i relacji w bazie. Możemy przykładowo zażądać sporządzenia raportu zawierającego zestawienie i charakterystykę wszystkich obiektów z modeli określonego typu lub tylko wybranego modelu, lub też atrybutów dla każdego obiektu.

Raport może być sporządzony w samym ARIS-Toolset (własny edytor umożliwia redagowanie raportu), lub w edytorze Word for Windows.

Relacje między obiektami mogą być zestawione w postaci tabelarycznej (macierzowej). Program może także sporządzić tablicę zawierającą wartości atrybutów dla każdego obiektu (z modelu, modeli określonego typu, grupy itp.). Wyniki o postaci tabelarycznej mogą zostać automatycznie wstawione do arkusza kalkulacyjnego Excel.

Wykorzystywanie modeli referencyjnych

Przy wykorzystaniu architektury ARIS w firmie IDS Prof. Scheer GmbH opracowano wiele modeli – wzorców, odnoszących się do poszczególnych branż przemysłowych, m.in. energetycznego, chemicznego, farmaceutycznego, a także przedsiębiorstw handlowych, bankowych i ubezpieczeniowych. Noszą one nazwę modeli referencyjnych [10]. Na podstawie tych modeli, stosownie do wymagań zdefiniowanych przez użytkownika w postaci reguł podanych programowi, ARIS-Toolset generuje modele funkcji, danych, procesów i organizacji dostosowane do potrzeb konkretnego przedsiębiorstwa. Użytkownik może także kopiować modele referencyjne i bezpośrednio wprowadzać odpowiadające mu modyfikacje.

ARIS-Toolset może przeprowadzić analizę porównawczą modelu użytkownika z modelem referencyjnym i wyspecyfikować różnice między obu modelami.

Niemiecka firma *software'owa* SAP AG wykorzystuje ARIS-Toolset do opisu oferowanego przez siebie oprogramowania – zintegrowanego systemu zarządzania przedsiębiorstwem R/3[2]. Posiadany opis R/3 w konwencji model ARIS ułatwia wdrażanie tego systemu w przedsiębiorstwach: przede wszystkim dzięki możliwości sprawnego wygenerowania modeli, które są dopasowane do rzeczywistych potrzeb konkretnego przedsiębiorstwa, co pozwala na parametryzację (ang. *customizing*) i wygenerowanie dobrze dopasowanego oprogramowania [4].

[2] 4600 instalacji w 41 krajach; 36% udział w rynku zintegrowanego oprogramowania dla zarządzania przedsiębiorstwami (wg danych GartnerGroup).

Tryby pracy

ARIS-Toolset udostępnia trzy tryby pracy: modelowania (ARIS **Modeller**), analizy (ARIS **Analyzer**) i nawigacji (ARIS **Navigator**).

ARIS Modeller umożliwia budowanie modeli danych, funkcji, organizacji i sterowania. Modeller zapewnia sprawne budowanie modeli, udostępniając użytkownikowi oprócz narzędzi do łatwego wprowadzania obiektów i relacji między nimi, wiele dodatkowych możliwości, jak: przeglądarkę dla szybkiego przemieszczania się z jednej części diagramu do innej, dziedziczenie atrybutów, utworzenie modelu powiązanego z wskazanym obiektem, tworzenie powiązania między bieżącym obiektem i istniejącym modelem, uruchamianie programów przypisywanych obiektom jako ich atrybuty, usuwanie niejednoznaczności w definicjach obiektów, wyszukiwanie obiektów, zarządzanie obiektami i modelami w grupach (np. przenoszenie modeli z grupy do innej grupy). **Modeller** zapewnia kontrolę spójności podczas modelowania, np. sygnalizuje próby wprowadzenia obiektu o tej samej nazwie, która już istnieje w bazie.

ARIS Analyzer umożliwia sporządzanie analiz, ułatwiających optymalizację systemu, a także generowanie – na podstawie modeli referencyjnych – modeli dostosowanych do potrzeb konkretnych przedsiębiorstw. Analizy mogą dotyczyć czasu, kosztów i prawdopodobieństw realizacji procesów i podprocesów. Dodatkowe funkcje umożliwiają szybkie wyszukiwanie interesujących nas modeli, obiektów, czy też relacji, np. możemy zestawić wszystkie relacje, jakie ma wskazany przez nas obiekt z innymi obiektami w bazie. Generowanie modeli na podstawie modeli referencyjnych pozwala wykorzystać już nagromadzoną wiedzę o strukturach i procesach.

ARIS Navigator ułatwia poruszanie się w systemie zbudowanych modeli. Możemy – przykładowo – dla danego obiektu z modelu danych odszukać funkcje, które korzystają z tego obiektu danych, znaleźć jednostki organizacyjne odpowiedzialne za realizację danej funkcji, lub też ustalić, za jakie obiekty danych odpowiedzialna jest dana jednostka organizacyjna. Navigator pozwala nam także znaleźć modele, które zawierają wystąpienia obiektów określonego typu, ukryć obiekty na diagramie, wyszukać obiekty powiązane bezpośrednio z wskazanym obiektem, wyszukać modele zawierające kopie wskazanego obiektu itd.

Warunki użytkowania

Oprogramowanie ARIS-Toolset jest oferowane w trzech wersjach językowych: niemieckiej, angielskiej i francuskiej. Obecnie wykorzystujemy angielskojęzyczną wersję 2.1 tego programu. Ta wersja programu jest przeznaczona do eksploatacji na komputerach z procesorem co najmniej 386 (zalecane 486). Wymaganych jest 10 MB pamięci dyskowej oraz o najmniej 8 MB RAM (zalecane 16 MB); system Windows 3.1.

* * *

Przydatność ARIS-Toolset dla BPR wynika z faktu, że program ten pozwala na kompletne i spójne przedstawianie procesów ekonomicznych w organizacji. W modelach są ujmowane: zdarzenia, funkcje, jednostki organizacyjne, obiekty danych, systemy aplikacji, nośniki danych. Procesy mogą być uszczegóławiane za pomocą dowolnej liczby poziomów hierarchii. Dzięki kompletności i spójności w ujmowaniu procesów, zarówno podczas analizy istniejących, jak i podczas projektowania nowych procesów, są natychmiast wykrywane takie usterki, jak: nieadekwatny lub niewystarczający stopień wspomagania funkcji przez techniki informatyczne, nieefektywna kolejność realizacji funkcji, częsta zmiana środków przechowywania informacji, częsta zmiana wydziałów podczas realizacji procesu przetwarzania informacji, redundantne wykonywanie funkcji.

Procesy są opisywane na różnych poziomach abstrakcji, co ułatwia komunikację między różnymi grupami ludzi zaangażowanych w BPR: menedżerowie, ekonomiści, technicy, informatycy. Szczególnie istotne jest zapewnienie związku ze sferą zastosowań technologii informatycznych i komunikacyjnych. BPR wymaga wprowadzania nowych rozwiązań informatycznych. Po rekonstrukcji procesów istniejące systemy informatyczne muszą być bowiem zazwyczaj zastąpione przez nowe. Przyczyną jest fakt, że „stare" systemy nie są już w stanie wspomagać „nowych" procesów wystarczająco efektywnie. Z kolei wdrażanie nowych, lecz coraz bardziej „wyrafinowanych" rozwiązań informatycznych, np. systemu w architekturze klient-serwer zamiast systemu opartego na scentralizowanej bazie danych, jest coraz bardziej złożone, pracochłonne i kosztowne. ARIS wspomaga przeprowadzanie customizingu, co ułatwia wdrażanie gotowych, standardowych systemów, zawierajcych te nowe rozwiązania i pozwala na zredukowanie nakładów pracy i kosztów na ich wdrożenie.

Ujmując te same procesy na różnych poziomach abstrakcji: użytkownika (ang. *Business Concept*), specyfikacji projektowej (ang. *Data Processing Concept*), technicznej implementacji (ang. *Implementation*) ARIS-Toolset ułatwia planowanie strategii informacyjnej powiązanej z ogólną strategię biznesową firmy. Możliwe jest wskazanie stopnia dopasowania istniejącej praktyki przetwarzania informacji do projektowanych procesów. Modele ARIS ukazują w szczególności: jakie bazy danych i aplikacje są wykorzystywane i na jakim sprzęcie, jakie aplikacje pracują z jakimi bazami danych, a także jakie elementy systemu (interfejsy użytkownika, dane, funkcje) są zlokalizowane na jakim sprzęcie.

LITERATURA

[1] Aichele Ch., Kirsch J., Geschäftsprozeßanalyse auf Basis von Kennzahlensystemen, *m & c* 2/95, s. 123–132

[2] *ARIS-Toolset. Dokumentacja programu.* IDS Prof. Scheer GmbH, Saarbrücken 1994

[3] Baun H.-J., Fröschl F., Yalçin A.: Business Reengineering in der Telecom-Industrie. Best Practice Models, *m & c* 2/95, S. 139–147

[4] Bügers A., Wilken Th.: Unterstützung der SAP R/3-Einführung mit ARIS, *m & c* 3/94, S. 197–202

[5] *BPR, GartnerGroup Research Note.* July 1995, GartnerGroup Services 1995

[6] Hess Th., Brecht L.: *State of the Art des Business Process Redesign,* Gabler 1995

[7] *IDS Newsletter* No. 3, Aug./Sept. 1995

[8] Lasek M.: *Modelowanie procesów przetwarzania informacji w przedsiębiorstwie. Architektura zintegrowanych systemów informacyjnych ARIS.* Seria: Studia Informatyki Gospodarczej, Wyd. IWi Universität des Saarlandes w Saarbrücken – WNE Uniwersytet Warszawski, Warszawa, czerwiec 1995

dokończenie na s. 25

Zagadnienia pozyskiwania wiedzy dla sterowników rozmytych (1)

Jarosław Stańczak
Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Sytuacja, w której brak jest jednoznacznego określenia jakiegoś zjawiska lub pojęcia, jest bardzo częsta w świecie rzeczywistym. Człowiek na każdym prawie kroku styka się z informacją niepełną, niepewną lub nieprecyzyjną. Podejmowanie decyzji przy wykorzystaniu takich przesłanek nie jest dla ludzkiego mózgu czymś bardzo trudnym. Inaczej rzecz się ma z komputerami, których działanie jest oparte na logice klasycznej, gdzie wszystkie fakty i zdania mogą być jedynie prawdziwe lub fałszywe. Logika rozmyta jest narzędziem, które umożliwia łatwe prowadzenie działań na informacji nieprecyzyjnej (*rozmytej*), dostosowanej do reprezentacji komputerowej. Pozwala to opisywać sytuacje, dla których trudno było stworzyć modele za pomocą klasycznego aparatu matematycznego.

Logika rozmyta (ang. *fuzzy logic*) jest uogólnieniem klasycznej logiki zero-jedynkowej dla zdań i reguł, których nie można z całą pewnością określić mianem prawdziwych lub fałszywych. Jest to jedno z popularnych obecnie rozwiązań umożliwiających wnioskowanie przy wykorzystaniu informacji nieprecyzyjnej lub przybliżonej. Ułatwia ono podejmowanie decyzji w sytuacjach związanych z nieokreślonością zarówno przesłanek, jak i reguł decyzyjnych.

Wnioskowanie w logice rozmytej jest przeprowadzane zazwyczaj z wykorzystaniem dobrze znanej z logiki klasycznej reguły *modus ponens*, która prowadzi do powstania reguł decyzyjnych o postaci *„jeśli... to..."* (ang. *if... then...*). Wartości logiczne takich reguł mogą zawierać się w przedziale domkniętym $\langle 0, 1 \rangle$, czyli na ogół **są prawdziwe lub fałszywe tylko w jakimś stopniu**. Przesłanki i konkluzje reguł opisują przynależność zmiennych decyzyjnych do określonych zbiorów rozmytych.

Mgr inż. Jarosław STAŃCZAK jest absolwentem Wydziału Elektroniki Politechniki Warszawskiej. W roku 1994 rozpoczął studia doktoranckie PW na Wydziale Elektroniki i Technik Informacyjnych. Jego zainteresowania dotyczą zagadnień sztucznej inteligencji, a w szczególności: układów rozmytych, uczenia się maszyn i sztucznych sieci neuronowych.

Pojęcie zbioru rozmytego (ang. *fuzzy set*) jest ściśle związane z logiką rozmytą. Jego cechą charakterystyczną jest to, że stopień przynależności danego elementu do zbioru określa się za pomocą pewnego współczynnika, wagi tego elementu. Z wag tych, określonych dla każdego elementu zbioru, powstaje funkcja, zwana funkcją przynależności do danego zbioru (ang. *membership function*), oznaczana często symbolem $m_A(x)$. Wartości tej funkcji zawarte są w przedziale $\langle 0, 2 \rangle$. Zerowa wartość funkcji przynależności odpowiada przypadkowi, kiedy x nie należy do zbioru A, z kolei dla $m_A(x) = 1$, wiemy, że x na pewno jest elementem zbioru A. Wartości pośrednie określają przypadki, co do których nie można się wypowiedzieć jednoznacznie.

Funkcja przynależności dla zbioru rozmytego jest określana najczęściej subiektywnie. Jej kształt jest dobierany do konkretnego problemu i nie jest w jakiś sposób narzucony. Funkcja przynależności musi jednakże spełniać następujące warunki: **monotoniczności** (im bardziej element „pasuje" do zbioru, tym większą wartość przyporządkowuje mu funkcja przynależności) i **symetrii** (elementy w równym stopniu spełniające kryteria przynależności do zbioru rozmytego, muszą mieć takie same wartości funkcji przynależności). Dokładny opis właściwości zbiorów rozmytych i wielu zagadnień z nimi związanych może czytelnik znaleźć np. w pracy [1].

Regulatory rozmyte

Jedną z dziedzin, gdzie systemy logiki rozmytej są często wykorzystywane, jest automatyka. Stosuje się tam do budowy sterowników układy oparte na logice rozmytej. Typową strukturę takiego regulatora przedstawia rys. 1. Układ taki jest właściwie systemem ekspertowym, którego zadaniem jest podejmowanie decyzji o wartościach sterowań na podstawie zmierzonych wartości wyjściowych z obiektu oraz wbudowanych reguł decyzyjnych.

Regulator oparty na wnioskowaniu rozmytym jest nieliniowym układem sterującym. Sterownik rozmyty umieszcza się w układzie sterowania w sposób tradycyjny, wykorzystując ujemne sprzężenie zwrotne (rys. 2a). Czasem stosuje się obok siebie dwa regulatory: tradycyjny i rozmyty, co często daje lepsze rezultaty niż zastosowanie tylko jednego z nich (rys. 2b). Polepszenie parametrów regulacji przy użyciu dwóch regulatorów polega np. na wykorzystaniu zerowego uchybu ustalonego regulatora PID, gdy tymczasem w regulatorach rozmytych występuje najczęściej niewielki uchyb ustalony lub oscylacje wokół punktu pracy. Regulator rozmyty daje natomiast znaczną poprawę właściwości dynamicznych układu.

Zasada działania sterownika rozmytego jest dość prosta. Wartość y na wejściu regulatora jest poddawana tzw. *fuzyfikacji*, czyli zamianie na *zmienną lingwistyczną*[1].

Rys. 1. Schemat regulatora rozmytego, x – sygnał wejściowy, u – sygnał sterujący; x_1, x_2 – zmienne lingwistyczne (odpowiednio wejściowa i sterująca); c_1, c_2 – współczynniki skalujące wartości wejściowe i sterowanie

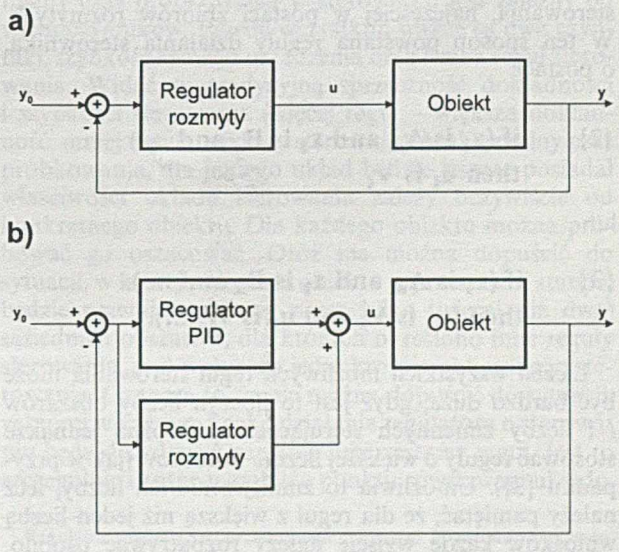
Polega to na znalezieniu zbioru rozmytego, do którego wartość ta należy oraz obliczeniu jej funkcji przynależności do tego zbioru. Procedura taka jest stosowana do wszystkich zmiennych wchodzących do regulatora. Następnie określany jest zestaw reguł aktywnych, czyli takich, które mają spełnione części warunkowe z niezerowymi wartościami funkcji przynależności. Dla każdej z reguł aktywnych obliczana jest jej waga. Najczęściej stosuje się w tym celu *operację minimum*[2]. Wagą danej reguły staje się najmniejsza z wartości funkcji przynależności przesłanek tej reguły. Następnie z zestawu reguł aktywnych eliminowane są te, które dla takich samych konkluzji mają najniższe wagi (*operacja maksimum*). Konkluzje otrzymanego w ten sposób zestawu reguł pokazują, do jakich zbiorów rozmytych powinny należeć sterowania. Aby znaleźć konkretne wartości sterowań, należy przeprowadzić tzw. *defuzyfikację* dla każdej zmiennej lingwistycznej, otrzymanej w procesie wnioskowania. Istnieje wiele sposobów defuzyfikacji (patrz np. [1]). Najczęściej stosowana jest tzw. *metoda środka ciężkości*, polegająca na obliczeniu środka ciężkości figury geometrycznej powstałej z zestawienia funkcji przynależności[3] do otrzymanych w procesie wnioskowania zbiorów rozmytych dla jednej wielkości sterującej.

Sterowniki, wykorzystujące ideę wnioskowania rozmytego, charakteryzują się więc prostotą działania. Szczególnie istotny jest tu fakt wykorzystywania w nich jedynie prostych działań matematycznych, bez konieczności całkowania, różniczkowania, czy obliczania wartości skomplikowanych funkcji. Ważną zaletą tych układów jest brak konieczności dobrej znajomości modelu obiektu w trakcie ich konstruowania. Nabiera to dużego znaczeia przy projektowaniu sterowników dla obiektów silnie nieliniowych lub obiektów z niepewnością (przez *niepewność* należy tu rozumieć zarówno niedokładną znajomość obiektu, jak i jego niedeterministyczne zachowanie się). W tradycyjnych metodach syntezy układów sterujących sprawą bardzo istotną była znajomość obiektu (model obiektu) lub modelu niepewności obiektu (model zakłóceń i sygnałów losowych, model losowego zachowania się obiektu). Przy syntezie sterownika rozmytego staje się istotna praktyczna wiedza o tym, jak należy obiektem sterować. Oczywiście wiedza ta nie oznacza braku konieczności uznania stanu obiektu, lecz reguły sterowania opierają się tu raczej na jakościowej niż ilościowej ocenie stanu obiektu.

Działanie sterownika opartego na wnioskowaniu rozmytym przypomina w dużym stopniu działanie człowieka, który dobrze sobie radzi ze sterowaniem nawet bardzo skomplikowanymi procesami robiąc to „na wyczucie". Wyczucie to najczęściej nie polega na dokładnej znajomości praw rządzących procesem, a raczej na doświadczeniu zdobytym przy posługiwaniu się tym obiektem. Istotna jest tu praktyczna wiedza o zachowaniu się obiektu, na skutek różnych działań operatora. Wiedza ta, zapisana w postaci odpowiednich reguł sterowania, staje się systemem ekspertowym o sterowaniu danym obiektem i może służyć za podstawę przy projektowaniu sterownika dla danego procesu. Reguły sterowania takiego systemu mają najczęściej charakter przybliżony, „rozmyty", a nie dokładny, dlatego też dobrym rozwiązaniem jest tu budowa sterownika rozmytego.

W ten sposób pojawiła się pierwsza metoda pozyskiwania reguł działania dla takiego sterownika – należy wykorzystać doświadczenie eksperta w danej dziedzinie. Wiedza ta powinna dać odpowiedź na następujące pytania:

– *Jak dobrać zakresy zmiennych charakteryzujących dany układ regulacji?*



Rys. 2. Struktura układu regulacji z regulatorem rozmytym
a) tylko z regulatorem rozmytym;
b) z regulatorem tradycyjnym (np. PID) i regulatorem rozmytym;
y_0 – wartość zadana regulatora, u – sterowanie, y – zmierzona wartość wyjściowa z obiektu.

[1] *Zmienna lingwistyczna* lub *etykieta lingwistyczna* – jest to właściwie nazwa zbioru rozmytego, do którego została zaklasyfikowana bieżąca wartość zmiennej y.

[2] Operacje **min** i **max** stosowane przy przetwarzaniu reguł w sterowniku rozmytym są przykadami odpowiednio t-normy i s-normy. Istnieje wiele innych przykładów norm wykorzystywanych we wnioskowaniu rozmytym. Szersze informacje na ten temat można uzyskać np. w pracy [1].

[3] Funkcje przynależności dla zmiennych sterujących otrzymane po procesie wnioskowania są modyfikowane tak, aby ich maksymalna wartość nie przekraczała wartości wagi swojej reguły. Można to uzyskać np. przez „przycięcie" funkcji do tej wartości lub też wymnożenie jej przez tę wagę.

- *Jak podzielić przestrzeń stanów na zbiory rozmyte?*
- *Jakie reguły sterowania zastosować?*
- *Jaki powinien być okres próbkowania stanu obiektu (czas między kolejnymi decyzjami o wartości sterowania)?*

Wykorzystanie wiedzy eksperta do tworzenia reguł sterowania

Najprostszym sposobem utworzenia zestawu reguł dla sterownika rozmytego jest wykorzystanie wiedzy człowieka, dobrze znającego dany problem. Rozwiązanie problemu polega na odpowiednim rozbiciu wybranej przestrzeni zmiennych stanu na obszary (ang. *clusters*), którym przyporządkowuje się określone sterowania. Podziału takiego dokonuje się dzieląc cały zakres zmienności każdej ze zmiennych (zarówno stanu, jak i sterujących) na przedziały. Każdy przedział staje się zbiorem rozmytym, dla którego określa się funkcję przynależności. Sposób podziału rozpatrywanego zakresu dla każdej ze zmiennych, liczba przedziałów (zbiorów rozmytych), na które została podzielona oraz dobór funkcji przynależności do nich zależy tu od wiedzy eksperta i ma znaczny wpływ na późniejsze zachowanie się obiektu. Najczęściej stosowany jest jednolity[4] podział rozpatrywanego wycinka przestrzeni stanów na hipersześciany powstające na przecięciach obszarów należących do różnych zbiorów rozmytych, na które podzielono zmienne stanu. Otrzymuje się w ten sposób obszary, których liczba wyraża się wzorem:

$$\{1\} \qquad l = \prod_{i=1}^{i=n} k_i$$

gdzie:
l – liczba obszarów; k – liczba przedziałów (zbiorów rozmytych) dla i-tej zmiennej; n-liczba zmiennych stanu.

Dla każdego obszaru należy teraz określić właściwe sterowania, najczęściej w postaci zbiorów rozmytych. W ten sposób powstaną reguły działania sterownika, o postaci:

$$\{2\} \qquad \textbf{if } (x_1 \textbf{ is } A_m \textbf{ and } x_2 \textbf{ is } B_n \textbf{ and } \ldots) \textbf{ then } u_1 \textbf{ is } V_s$$

lub

$$\{3\} \qquad \textbf{if } (x_1 \textbf{ is } A_m \textbf{ and } x_2 \textbf{ is } B_n \textbf{ and } \ldots) \textbf{ then } (u_1 \textbf{ is } V_s \textbf{ and } u_2 \textbf{ is } W_r \ldots).$$

Liczba wszystkich możliwych reguł sterowania może być bardzo duża, gdyż jest to iloczyn liczby obszarów l i liczby zmiennych sterujących n. Można jednakże stosować reguły o większej liczbie wniosków (jak w przypadku {3}). Umożliwia to zmniejszenie ich liczby, lecz należy pamiętać, że dla reguł z większą niż jeden liczbą wniosków każde wyjście należy rozpatrywać osobno. Często zamiast podziału jednolitego stosuje się podział nieregularny. Możliwe jest to wtedy, gdy wiele sąsiednich obszarów wyróżnionych np. przez podział jednolity ma takie same sterowania. Obszary takie są łączone w większe rejony i w rezultacie powstaje z nich tylko jedna reguła.

Optymalny dobór reguł, ich liczby, a także parametrów funkcji przynależności, to zagadnienia decydujące o jakości sterowania. Najważniejsze znaczenie ma tu oczywiście odpowiedni dobór części wynikowej reguł. Przy nieodpowiednim doborze układ może stać się niestabilny. Podobne skutki może dać także zbyt mała liczba zbiorów rozmytych dla zmiennych lub ich niewłaściwe rozgraniczenie. Rozwiązanie problemu doboru reguł i parametrów jest w dużej mierze zależne od konkretnego zadania sterowania i trudno jest znaleźć ogólną metodę ich dobierania. Nawet ekspertowi w dziedzinie sterowania jakimś obiektem dość trudno jest zamienić swoją wiedzę na odpowiednie reguły dla regulatora rozmytego.

Należy zauważyć, że o ile ekspertowi nie sprawia dużych trudności określenie sterowań dla odpowiednich obszarów w przestrzeni stanów, to jednak często okazuje się, że nie można tej wiedzy bezpośrednio zamienić na formuły dla regulatora rozmytego. W wielu przypadkach otrzymuje się zestawy reguł, które powodują powstawanie w przestrzeni stanów rejonów, w których układ zachowuje się nie tak, jak można by tego oczekiwać[5]. Można czasem zaobserwować zjawisko polegające na tym, że dla pewnych warunków początkowych sterowany obiekt staje w martwym punkcie, wpada w oscylacje lub zaczyna poruszać się po bardzo nieoptymalnych trajektoriach. Przyczyną takiego zachowania układu jest często zła współpraca sąsiadujących reguł, spowodowana ich symetrią lub nieodpowiednim zakresem działania. Można temu zjawisku przeciwdziałać na kilka sposobów:

- *stosując reguły zaburzające symetrię (wytrącające układ z niepożądanego położenia), działające tylko w tych obszarach;*
- *projektując zestaw reguł tak, aby nie sąsiadowały ze sobą (nie zachodziły na siebie funkcje przynależności) w przestrzeni stanów obszary, które nadają przeciwne wartości jakiejś zmiennej sterującej;*
- *zachowując dla sterowań w całej rozpatrywanej części przestrzeni stanów pewną tendencję, powodującą „ześlizgiwanie się" obiektu do pożądanego punktu pracy.*

O ile sposób pierwszy i drugi nie wymagają specjalnych komentarzy, to do przypadku trzeciego należy dodać kilka słów wyjaśnienia. Punkty siodłowe w przestrzeni stanów pojawiają się w takich miejscach, gdzie z jednej ich strony należy sterować w jedną stronę, a z drugiej w drugą. Takie sterowania wydają się optymalne. Często jednak można zastosować gorsze zestawy reguł, dające taki sam charakter sterowań po obu stronach krytycznego punktu, powodujące przejście układu do punktu stacjonarnego po dłuższej trajektorii, lecz bez tworzenia niepożądanych „zawirowań" w hiperpowierzchni określającej sterowania obiektu.

Zdecydowanie, które z wymienionych reguł zastosować w danym przypadku – jest dla projektanta systemu trudnym zadaniem do rozwiązania.

W wypadku sterowania niezbyt skomplikowanym układem dynamicznym, Macvicar i Whelen podali przy-

[4] *Jednolity*, czyli tworzymy obszar dla każdej kombinacji zbiorów rozmytych dla poszczególnych zmiennych. W przypadku dwuwymiarowym przypomina on kratę. Nie oznacza to jednak, że wydzielone regiony muszą mieć jednakowe rozmiary.

[5] Są to tzw. *punkty siodłowe*, które powstają np. wtedy, gdy w otoczeniu takiego punktu dwie reguły generują przeciwne co do znaku i równe co do wartości sterowania.

kład reguł decyzyjnych i podziału przestrzeni stanów. Regulator oparty na takich regułach zachowuje się podobnie do klasycznego regulatora PD. Przy projektowaniu regulatora rozmytego zakładamy, że znamy wartość uchybu regulacji – **e** i pochodnej uchybu regulacji – **de**. Na podstawie tych danych oraz wbudowanych reguł układ regulatora rozmytego generuje sterowanie (u) minimalizujące uchyb regulacji. Reguły przedstawia tabela 1. Jest to oczywiście skrótowy opis reguł, wykorzystujący symboliczne nazwy zbiorów rozmytych. Właściwe reguły działania regulatora można na podstawie tabeli zapisać np. tak:

JEŻELI **((de ∈ UD) ∧(e ∈ DD)** ***TO*** **(u∈UZ)**

Parametry funkcji przynależności do podanych w niej zbiorów rozmytych należy dobrać stosownie do rozpatrywanego zadania sterowania.

Macierz Macvicara i Whelena

e \ de	UD	US	UM	UZ	DZ	DM	DS	DD
DD	UZ	UM	US	UD	UD	UD	UD	UD
DS	DZ	UZ	UM	US	US	US	UD	UD
DM	DM	DM	UZ	UM	UM	UM	US	UD
DZ	DS	DS	DM	DZ	UZ	UM	US	US
UZ	DS	DS	DM	DZ	UZ	UM	US	US
UM	DD	DS	DM	DM	DM	DZ	UM	UM
US	DD	DD	DS	DS	DS	DM	DZ	DZ
UD	DD	DD	DD	DD	DD	DS	DM	DZ

gdzie:
e – błąd regulacji; de – pochodna błędu regulacji;
Zbiory rozmyte:
dla e, de i wyjścia: {DD, DS, DM, DZ, UZ, UM, US, UD};
DD – „dodatnie duże", DS – „dodatnie średnie", DM – „dodatnie małe", DZ – „dodatnie zero",
UD – „ujemne duże", US – „ujemne średnie", UM – „ujemne małe", UZ – „ujemne zero".

Sposób fragmentacji przestrzeni stanów ma istotne znaczenie dla działania regulatora rozmytego. Wiąże się to ściśle z kształtami i liczbą zbiorów rozmytych dla zmiennych stanu. Parametry te zależą w znacznej mierze od charakteru rozwiązywanego problemu. Z reguły nie stosuje się mniej niż trzech przedziałów fuzyfikacji dla zmiennych stanu (np.: ujemne, zero, dodatnie lub też: małe, duże, średnie). Im więcej zbiorów rozmytych powstanie, tym większa będzie liczba reguł, a co za tym idzie lepsza dokładność reakcji na zmiany zachodzące w danym obiekcie. Z drugiej strony zwiększą się trudności związane z właściwym doborem parametrów reguł, a także większy będzie nakład obliczeń w trakcie pracy układu. Dla zbyt małej liczby przedziałów fuzyfikacji parametry regulatora mogą się znacznie pogorszyć, aż do otrzymania niestabilnego układu regulacji. Zmienne sterujące nie stanowią tu dużego problemu, często zamiast lingwistycznych wartości wyjść stosuje się singletony rozmyte[6]. Liczba zbiorów rozmytych lub singletonów może być duża i powinna być wystarczająca dla danego problemu, gdyż nie wpływa ona w decydujący sposób na liczbę reguł dla regulatora.

[6] Singletonem rozmytym nazywa się konkretną wartość liczbową zastosowaną jako konkluzję reguły.

Ze sposobem fragmentacji przestrzeni stanów są związane parametry funkcji przynależności zmiennych do zbiorów rozmytych. W przeważającej liczbie przypadków stosuje się funkcje o kształcie trapezoidalnym lub trójkątnym (traktując trójkąt jako szczególny przypadek trapezu) z uwagi na prostotę obliczeń i dobre rezultaty otrzymywane przy ich wykorzystaniu. Mogą tu być wykorzystane także inne funkcje, spełniające postulaty wymienione we wstępie niniejszej pracy, np. w realizacjach cyfrowych trójkąt czy trapez zastępuje się odpowiednio funkcjami schodkowymi o pewnej liczbie poziomów dyskretyzacji.

Określenie parametrów funkcji przynależności o postaci trójkąta lub trapezu polega na znalezieniu czterech parametrów, będących współrzędnymi czterech wierzchołków trapezu (dla funkcji trójkątnej daną określającą wierzchołek trzeba zapamiętać dwukrotnie). Do określenia tych parametrów można podać jedynie dość ogólne właściwości, gdyż bardzo silnie zależą one od charakteru rozpatrywanego problemu. Dla obszarów bliskich celowi sterowania stosuje się funkcje przynależności o kształcie trójkątnym, o dość wąskiej podstawie. Zapewnia to dużą precyzję sterowania w pobliżu wymaganego punktu pracy systemu. W obszarach znacznie oddalonych od celu wartości sterowań nie muszą być określane tak precyzyjnie, dlatego też stosuje się tu funkcje trapezoidalne lub trójkątne, o rozciągniętej podstawie. Oznacza to, że w miarę oddalania się od wymaganego punktu pracy obszary, na które podzielona jest przestrzeń stanów rosną, natomiast w jego pobliżu maleją, tak aby można było bardzo precyzyjnie określić położenie aktualnego stanu obiektu i na tej podstawie dokładnie określić sterowania.

Wpływ okresu próbkowania stanu obiektu przez regulator rozmyty[7] na parametry układu regulacji ma charakter zbliżony, jak w klasycznych sterownikach cyfrowych. Wnosi on dodatkowe opóźnienie w układzie, prowadzące do powstawnia oscylacji w układzie, a przy dużym opóźnieniu układ zaczyna poruszać się w przestrzeni stanów po zupełnie przypadkowych trajektoriach. Minimalny czas próbkowania w sterowniku jest zależny od wielu czynników. Przede wszystkim ważne są dwa czynniki: szybkość samego urządzenia oraz liczba reguł sterowania. Widać tu tradycyjną sprzeczność dokładności i szybkości sterowania (więcej reguł – większa dokładność, mniej reguł – szybszy sterownik). Maksymalny czas próbkowania, dla jakiego układ będzie jeszcze posiadał właściwości układu sterowania zależy oczywiście od konkretnego obiektu. Dla każdego obiektu można próbować go oszacować. Otóż nie można dopuścić do sytuacji, w której położenie obiektu w przestrzeni stanów będzie przemieszczać się przez kilka (więcej niż dwa) sąsiednich obszarów, dla których określono inne reguły sterowania, pod wpływem jednokrotnie obliczonego sterowania. Oznacza to, że nie można dopuścić do sytuacji, w której w każdym kroku działania regulatora o sterowaniu będą decydować zupełnie inne zestawy reguł. Jest to szczególnie istotne w pobliżu punktu równowagi układu.

[7] Regulator rozmyty jest urządzeniem cyfrowym. Nowe wartości sterowań mogą być wyliczane tylko w określonych chwilach czasowych. Fakt ten wynika z samej zasady działania regulatora – wnioskowanie jest procesem, który trwa. Dlatego też okres próbkowania nie może być krótszy od czasu przetwarzania reguł.

dokończenie na s. 25

Integracja zarządzania sieciami IPX/SPX i TCP/IP (2)

Tomasz Lis, Artur Sobkowiak, Maciej Stroiński
Instytut Informatyki
Politechnika Poznańska

Instytut Informatyki Politechniki Poznańskiej ma wielosegmentową, wieloprotokołową, kręgosłupową sieć lokalną, do której za pośrednictwem wewnętrznych mostów międzysieciowych (ang. *internal router*) są dołączone sieci lokalne o architekturze NetWare, Internet (SUNOS, SCO, Unix, NetWare) i OSI (SunLink OSI), w których pracuje kilkadziesiąt mikrokomputerów. Wraz z rozwojem tych sieci staje się niezbędne posiadanie systemów zarządzania tymi sieciami, w szczególności systemu zarządzania siecią NetWare.

Założenia projektowe systemu zarządzania

System operacyjny stanowisk usług plikowych sieci NetWare udostępnia bardzo ograniczony zestaw danych, które obrazują zachowanie samego stanowiska usługowego (wykorzystanie pamięci, procesora, dysków itp.), lecz nie pozwalają na ocenę pracy całej sieci. Poza tym nie ma możliwości archiwizacji prezentowanych danych.

Istnienie oddzielnych systemów zarządzania dla poszczególnych sieci wskazuje na konieczność ich itegracji. Z tego względu wymaganiem koniecznym dla systemu zarządzania sieciami NetWare jest możliwość jego współpracy z dowolnym systemem nadrzędnym, np. stosującym protokół CMIP lub SNMP.

Dostępne oprogramowanie nie zapewniało takich możliwości i dlatego zdecydowano się na zaprojektowanie własnego systemu zarządzania.

Przed systemem zbierania informacji w sieci NetWare postawiono następujące wymagania funkcjonalne:

- system powinien pracować na stanowisku usługowym sieci lub na stacji roboczej tej sieci,
- system powinien umożliwiać uzyskanie informacji o konfiguracji sieci oraz typie i aktywności dowolnych stanowisk sieci, również tych znajdujących się w innych segmentach,
- system powinien umożliwiać ciągłe monitorowanie wszystkich stanowisk sieci i gromadzenie (archiwizację) wybranych danych o ich pracy,
- system powinien przetwarzać uzyskiwane dane w celu otrzymania zbiorczej informacji o pracy całej sieci; informacje te powinny być archiwizowane,
- system powinien umożliwiać przekazywanie uzyskanych danych do nadrzędnego systemu zarządzania za pomocą protokołu SNMP lub CMIP, czyli realizować funkcję agenta zarządzania dla zarządcy wyższego poziomu,
- brak konieczności modyfikowania oprogramowania stanowisk usługowych i stacji roboczych sieci NetWare.

Prezentacja informacji, a w szczególności jej przesyłanie przez protokoły zarządzania, wymaga precyzyjnego określenia jej syntaktyki i semantyki (formatu i znaczenia). Powszechnie stosowaną metodę stanowi definiowanie bazy informacji zarządzania MIB. W literaturze było brak propozycji definicji obiektów związanych z zarządzaniem w sieci NetWare i z protokołami IPX/SPX. Ze względu na specyfikę dostępnych informacji nie jest też możliwe bezpośrednie zastosowanie definicji dostępnych dla innych sieci np. MIB-II dla sieci TCP/IP. W związku z tym powstała potrzeba stworzenia formalnej definicji obiektów opisujących sieć NetWare i stanowiących odniesienie dla projektowanego systemu zarządzania.

Przedstawiona dalej baza informacji zarządzania została zaprojektowana pod kątem protokołu SNMP. Jest ona w istocie rozszerzeniem MIB-II dla sieci TCP/IP. Zaproponowane definicje są zgodne z zasadami zawartymi w dokumencie RFC 1155 [10]. Takie podejście powinno zapewnić łatwą integrację projektowanego systemu z systemami zarządzania w sieciach TCP/IP, stosujących protokół SNMP.

Baza informacji zarządzania dla sieci NetWare

W bazie informacji zarządzania dla sieci NetWare – *NetWare-MIB* wyróżniono następujące grupy obiektów: *node, interfaces, ipx, spx, shell, bridge, network, netConfig, networkHistory, nodeHistory*.

Grupa *node* zawiera podstawowe informacje o wszystkich stacjach w sieci NetWare. Dane dla poszczególnych stacji są zawarte w tablicy nodeTable.

Grupa *interfaces* zawiera informacje o konfiguracji interfejsów sieciowych i ich statystykach. Dane te są zawarte w tablicy ifTable.

W grupie *ipx* zgromadzono statystyki modułów protokołu IPX poszczególnych stacji. Dane te są zawarte w tablicy ipxTable.

Grupa *spx* zawiera statystyki modułów protokołu SPX poszczególnych stacji. Dane te są zawarte w tablicy spxTable.

Grupa *shell* zawiera informacje o pracy powłok systemów operacyjnych stacji roboczych. Statystyki są zgromadzone w tablicy shellTable.

Grupa *bridge* zawiera informacje o ruchu międzysieciowym w węzłach międzysieciowych. Dane te są zawarte w tablicy bridgeTable.

Przedstawiony w pracy system zarządzania zrealizowano w ramach grantu KBN 335209102

Grupa *network* zawiera informacje o segmentach sieci i statystyki tych segmentów. Dane zgromadzone są w tablicy networkTable.

Grupa *netConfig* zawiera informacje o stacjach przyłączonych do poszczególnych segmentów. Przyporządkowanie to znajduje się w tablicy netConfigTable.

Grupa *networkHistory* umożliwia gromadzenie danych o pracy wybranych segmentów sieci. Informacje te są gromadzone w tablicy networkHistory na podstawie informacji kontrolnych zawartych w tablicy networkHistoryControlTable.

Grupa *nodeHistory* umożliwia gromadzenie danych o pracy wybranych stacji w sieci. Statystyki są gromadzone w tablicy nodeHistoryTable na podstawie informacji kontrolnych zawartych w tablicy nodeHistoryControlTable.

Na wydruku przedstawiono fragment formalnej definicji obiektów w *NetWare-MIB*.

```
Definitions

 NetWare-MIB DEFINITIONS ::= BEGIN

 IMPORTS
 enterprises, NetworkAddress, Counter,
 Gauge, TimeTicks
 FROM RFC1155-SMI
 DisplayString, PhysAddress FROM RFC1213-MIB-II
 OBJECT-TYPE FROM RFC1212;

 -This MIB module uses the extended OBJECT-TYPE macro
 -as defined in [9];

 pp OBJECT IDENTIFIER ::= {enterprises 1 }
 netware-mib OBJECT IDENTIFIER ::= {pp 1 }

 -textual conventions
□
 OwnerString ::=DisplayString
 Time ::=TimeTicks

 -groups in netware-mib

 node OBJECT IDENTIFIER ::= { netware-mib 1 }
 interfaces OBJECT IDENTIFIER ::= { netware-mib 2 }
 ipx OBJECT IDENTIFIER ::= { netware-mib 3 }
 spx OBJECT IDENTIFIER ::= { netware-mib 4 }
 shell OBJECT IDENTIFIER ::= { netware-mib 5 }
 bridge OBJECT IDENTIFIER ::= { netware-mib 6 }
 netAddress OBJECT IDENTIFIER ::= { netware-mib 7 }
 network OBJECT IDENTIFIER ::= { netware-mib 8 }
...
...
```

Implementacja systemu zarządzania w sieci NetWare

System realizujący założenia z punktu 5 nazwano **NetWare Stat**. Jego podstawowym przeznaczeniem jest automatyczne gromadzenie w plikach dyskowych informacji o pracy poszczególnych stacji i segmentów sieci NetWare, oraz udostępnianie zbieranych informacji nadrzędnym systemom zarządzania. System umożliwia pracę interakcyjną. Komunikacja z użytkownikiem odbywa się za pomocą systemu rozwijanych okien i menu zgodnego z systemem stosowanym we wszystkich standardowych narzędziach w sieci NetWare. Architektura systemu została przedstawiona na rysunku na następnej stronie.

System pracuje w dwóch trybach: interakcyjnym i automatycznego gromadzenia danych. Realizuje on następujące funkcje:

a) tryb interakcyjny
- prezentacja informacji o konfiguracji sieci (segmenty sieci, stacje w segmentach oraz ich aktywność),
- prezentacja konfiguracji aktywnych stacji,
- prezentacja statystyk poszczególnych komponentów stacji,

b) tryb automatycznego gromadzenia danych
- gromadzenie w plikach dyskowych informacji o czasie aktywności stacji i ich danych statystycznych,
- wyznaczanie uogólnionych danych reprezentujących działanie segmentu sieci jako całości i ich archiwizacja w plikach dyskowych,
- udostępnianie gromadzonych danych aplikacjom zarządzania przez protokół SNMP.

W obecnej wersji dane są rejestrowane co 10 minut. Wszystkie gromadzone dane są opatrzone etykietami czasowymi określającymi czas ich uzyskania.

Udostępniane przez system dane są pozyskiwane za pomocą usług diagnostycznych oferowanych przez sieć NetWare (biblioteka funkcji języka C [1]). Analiza dostępnych za pomocą tych usług danych pozwala zauważyć, że nie dają one kompletnego, zadowalającego obrazu aktywności stacji. Brak jest pewnych istotnych parametrów (np. liczby transmitowanych bajtów), inne są pogrupowane. Pomimo wspomnianych niedogodności implementacja systemu NetWare Stat musi korzystać z oferowanych przez sieć NetWare usług, ponieważ jest to jedyny sposób (przy założeniu nie modyfikowania standardowego oprogramowania NetWare) uzyskania informacji o pracy poszczególnych stacji w sieci.

System składa się z bazy obiektów zarządzania oraz modułów:
- zbierania danych,
- automatycznego gromadzenia danych,
- prezentacji danych,
- SNMP.

Baza obiektów zarządzania

Jądrem systemu jest baza obiektów zarządzania zawierająca dane o konfiguracji sieci oraz dane statystyczne poszczególnych elementów sieci. Jest ona zbudowana z zastosowaniem trzech klas obiektów: *NetNode*, *Network* i *InterNet*. Poszczególne moduły funkcjonalne są realizowane za pomocą odpowiednich metod tych obiektów.

Wystąpienia klasy *NetNode* reprezentują poszczególne stacje w sieci; wystąpienia klasy *Network* opisują segmenty sieci, a wystąpienie klasy *InterNet* reprezentuje całość połączonych segmentów sieci.

Konfiguracja sieci jest reprezentowana przez powiązania obiektów klas *NetNode*, *Network* i *InterNet*. Obiekty klasy *NetNode* tworzą listy reprezentowane przez obiekty

klasy *Network*. Obiekty klasy *Network* tworzą z kolei listę reprezentowaną przez obiekt klasy *InterNet*.

Obiekty klasy *NetNode* mają następujące atrybuty:

- fizyczny adres stacji (ang. *address*),
- logiczną nazwę stacji (ang. *name*), określoną w zbiorze konfiguracyjnym systemu,
- miejsce ustawienia stacji (ang. *location*), określone w zbiorze konfiguracyjnym,
- listę komponentów wchodzcych w skład stacji (ang. *configuration*),
- tablicę wskaźników do obiektów zawierających statystyki (ang. *stat*),
- flagę aktywności stacji (ang. *active*).

Obiekty klasy *Network* mają następujące atrybuty:

- adres IPX segmentu sieci (*netAddress*),
- logiczną nazwę segmentu sieci (*name*) (określona w zbiorze konfiguracyjnym),
- obiekt zawierający uogólnione statystyki segmentu sieci (*netStat*).

W celu ułatwienia użytkownikom identyfikacji stacji segmentów sieci wprowadzono możliwość nadawania im mnemonicznych nazw oraz określenia lokalizacji stacji. Przyporządkowanie nazw i lokalizacji stacjom o określonych adresach odbywa się w zbiorze konfiguracyjnym. Na przykład jeden z wierszy tego zbioru może wyglądać następująco:

```
00.80.00.13 00.c0.16.a7.d3.21
stacja_robocza_11 pokój_444
```

W wypadku braku nazwy jej rolę pełni adres stacji czy sieci. Zbiór konfiguracyjny umożliwia ponadto zdefiniowanie statycznej (zadeklarowanej przez operatora) konfiguracji sieci, na którą nakładana jest aktualna konfiguracja dynamiczna.

Pewnym problemem jest reprezentacja stacji mających więcej niż jeden interfejs sieciowy, np. węzłów międzysieciowych lub mostów. W przyjętym rozwiązaniu stacja taka jest reprezentowana przez obiekt klasy *NetNode* w każdym segmencie (reprezentowanym przez obiekt klasy *Network*), do którego jest przyłączona. Wyróżnionym obiektem jest ten związany z pierwszym interfejsem. Z nim związane są statystyki wszystkich komponentów stacji – poza statystykami pozostałych interfejsów, związanych z odpowiednimi obiektami w innych segmentach.

Dane statystyczne są reprezentowane przez atrybuty obiektów klas: *DriverStat, IPXStat, SPXStat, ShellStat, BridgeStat, NetStat*. Obiekty tych klas są związane z obiektami klas *NetNode* i *Network*.

Wspomniane atrybuty reprezentują wartości wybranych parametrów odpowiednich komponentów stacji lub całych segmentów sieci (klasa *NetStat*). Pamiętane są zawsze dwie wartości każdego parametru z bieżącego i poprzedniego pomiaru (nie dotyczy to klasy *NetStat*). Umożliwia to wyznaczenie przyrostu wartości danego parametru, w celu uzyskania wartości atrybutów obiektu klasy *NetStat* reprezentującego statystyki całego segmentu sieci. Poza wartościami parametrów są też pamiętane ich etykiety czasowe.

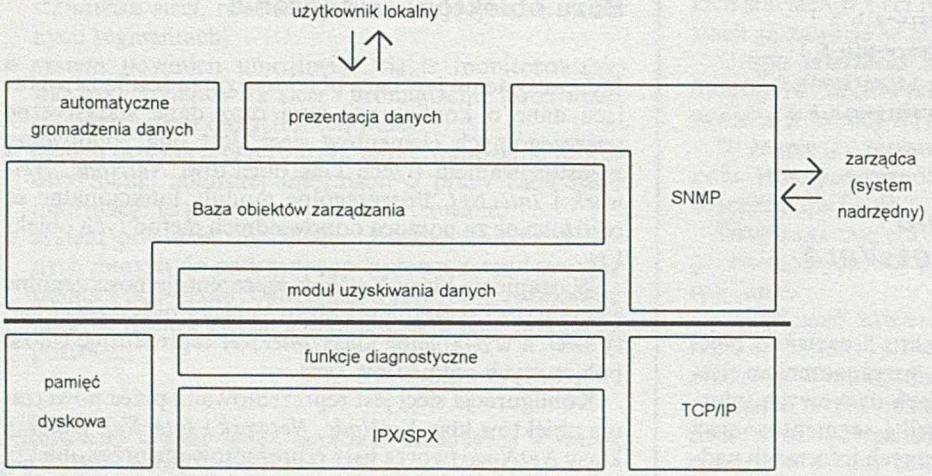
Obiekty *DriverStat, IPXStat, SPXStat, ShellStat, BridgeStat, NetStat* są wykorzystywane w trybie automatycznego gromadzenia danych. W trybie interakcyjnym informacje są uzyskiwane bezpośrednio z odpowiednich stacji.

Moduł zbierania danych

Moduł zbierania danych jest odpowiedzialny za dostarczanie danych o konfiguracji sieci i charakterystykach poszczególnych stacji. Moduł ten korzysta bezpośrednio z usług diagnostycznych oferowanych przez NetWare [6, 8].

Procedura sprawdzania konfiguracji sieci działa według następującego schematu:

1. Dla danego segmentu do każdej stacji w nim zainstalowanej jest wysyłane zapytanie o konfigurację (za pomocą datagramu rozgłoszeniowego). Dla każdej odpowiedzi jest powoływany obiekt klasy *NetNode*. Jest on umieszczany na liście odpowiedniego obiektu klasy *Network*. (Jeśli taki obiekt już istnieje, to stacja jest tylko oznaczana jako aktywna). Adres stacji jest dołączany do listy wykluczeń pakietu zapytania o konfigurację.
2. Krok 1. jest powtarzany tak długo, aż przestaną przychodzić odpowiedzi od nowych stacji.
3. Stacje mające wiele interfejsów sieciowych są traktowane w sposób specjalny – na podstawie odpowiedzi od nich są tworzone obiekty klasy *Network*. Następnie włącza się je na listę obiektu klasy *InterNet*. W ten sposób wykrywa się nowe segmenty sieci.



Architektura systemu NetWare Stat

4. Kroki 1–3 powtarza się dla każdego nowo wykrytego segmentu, aż zostanie rozpoznana konfiguracja całej sieci.

Moduł automatycznego gromadzenia danych

Moduł automatycznego gromadzenia danych jest odpowiedzialny za cykliczne uaktualnianie informacji o konfiguracji sieci, statystykach poszczególnych stacji i segmentów sieci. Dane te są archiwizowane w plikach dyskowych. Możliwe jest zabezpieczenie konsoli hasłem. Moduł działa według następującego algorytmu:

W ustalonych odstępach czasu (np. 10 min.) moduł jest uaktywniany i wykonuje cykl gromadzenia danych. W danym cyklu dla każdego segmentu są wykonywane następujące czynności:

1. Uaktualniana jest informacja o konfiguracji segmentu.
2. Z każdą aktywną stacją jest nawiązywane połączenie SPX, uzyskiwane są dane statystyczne i uaktualniane są obiekty statystyczne związane z daną stacją.
3. Na podstawie uaktualnionych danych poszczególnych stacji są wyliczane statystyki segmentu sieci i są uaktualniane atrybuty odpowiedniego obiektu klasy *NetStat*.
4. Dla każdej stacji są zapisywane w odpowiednim zbiorze aktualne wartości statystyk i czas ich uzyskania.
5. Zapamiętywane są statystyki segmentu sieci i czas ich uzyskania.

Moduł prezentacji danych

Moduł prezentacji umożliwia użytkownikowi pracę interakcyjną. Pozwala na prezentację informacji za pomocą systemu okien i menu stylizowanego na standardowe narzędzia NetWare. W tym trybie na konsoli jest prezentowana konfiguracja sieci (w postaci listy segmentów sieci i listy stacji w ramach danego segmentu) i jest możliwe wyświetlenie informacji o aktualnej konfiguracji i statystykach wybranej stacji.

Moduł SNMP

Moduł SNMP zapewnia dostęp do informacji zarządzania dla nadrzędnego systemu zarządzania, umożliwiając integrację systemu **NetWare Stat** z systemami zarządzania sieciami TCP/IP. Zawiera on implementację protokołu SNMP. Jego zadaniem jest komunikacja z nadrzędnym systemem zarządzania i interpretacja zapytań SNMP, polegająca na odwzorowaniu zmiennych z *NetWare MIB* w dane pozyskiwane przez moduł zbierania danych.

Uwagi implementacyjne

Program został napisany w języku C++ (w systemie Turbo C++). Korzysta on z bibliotek klas dostarczanych z systemem Turbo C++ oraz bibliotek „NetWare C Interface DOS" [6].

Uruchomienie programu jest możliwe na dowolnym stanowisku roboczym sieci NetWare wyposażonym w system operacyjny DOS. Wymagana pojemność pamięci jest zależna od wielkości monitorowanej sieci (dla sieci rzędu 50 stanowisk 640 K w zupełności wystarcza). W wypadku automatycznego monitorowania sieci niezbędne jest zapewnienie odpowiedniej pojemności pamięci dyskowej dla gromadzenia danych. Niezbędna pojemność dysku jest uzależniona od ilości monitorowanych węzłów i czasu monitorowania. Wymagane jest również zainstalowanie oprogramowania TCP/IP PC-NES v.4.0 firmy Sun Microsystems.

* * *

Prototypowe wersje przedstawionego systemu powstały już w czasie, gdy zaawansowane komercyjne produkty zarządzania siecią NetWare nie były jeszcze dostępne. Z konieczności system korzysta zatem tylko z tych mechanizmów pozyskiwania danych, jakie są standardowo dostępne w sieci NetWare. Ogranicza to oczywiście w pewnym stopniu jego możliwości, ale z drugiej strony nie wymaga ingerencji w konfigurację istniejących sieci, niezbędną np. w wypadku systemu NetWare Management System. Ponadto prezentowany system nie wymaga instalowania protokołów TCP/IP (poza stacją, na której system pracuje – w wypadku integracji z nadrzędnym systemem zarządzania). Istotną cechą prezentowanego systemu jest także możliwość współpracy z dowolnym nadrzędnym systemem zarządzania z protokołem SNMP. W tym wypadku system pracuje w roli agenta (proxy) zarządzania dla systemu nadrzędnego, udostępniając informacje o całej sieci NetWare. Dzięki takiemu podejściu, w pewnym sensie alternatywnemu do rozwiązania w NetWare Management System, proponowany system może być (pomijając kwestie finansowe) interesujący, szczególnie w wypadku gdy sieć NetWare stanowi tylko fragment infrastruktury sieciowej.

Przedstawiony system zrealizowano w Instytucie Informatyki Politechniki Poznańskiej, gdzie stanowi obecnie element systemu zarządzania siecią instytutową.

LITERATURA

[1] Case J.D., Fedor M., Schoffstall M.L., Davin C.: Simple NetWork Management Protocol (SNMP), RFC 1157, May 1990

[2] Lis T.: Koncepcja systemu do zbierania informacji w sieci NetWare. Raport IInPP RAP /93

[3] Lis T., Stroiński M.: System do zbierania danych statystycznych o pracy sieci komputerowej NetWare. Plakat na SET'94.

[4] McCloghrie K. (ed.): Extensions to the Generic-Interface MIB, RFC 1229, May 1991

[5] McCloghrie K., Rose M.T. (eds.): Management Information Base for Network Management of TCP/IP-bases Internets: MIB-II, RFC 1213, March 1991

[6] NetWare C Interface DOS

[7] NetWare System Calls DOS

[8] NetWare System Interface Technical Overview

[9] Rose M., McCloghrie K.: Structure and Identification of Management Information for TCP/IP-based Internets, RFC 1155, May 1990

[10] Rose M., McCloghrie K. (eds.): Concise MIB Definitions, RFC 1212, March 1991

[11] Stroiński M., Lis T., Sobkowiak A.: Integracja Zarządzania sieciami NetWare i TCP/IP. Raport IInPP RB-94/024

[12] Waldbusser S.: Remote Network Monitoring Management Information Base, RFC 1271, November 1991.

Programowanie w języku systemów eksperckich CLIPS (4)

Integracja systemów eksperckich z programami opracowanymi w innym środowisku

Tomasz Pytlak

Samodzielna Pracownia Sztucznej Inteligencji
Instytut Podstaw Elektroniki
Politechnika Warszawska

W dotychczas prezentowanych opisach języka CLIPS zostały przedstawione zasady tworzenia systemów eksperckich w tym języku. W poniższej pracy zostanie przedstawiony proces integracji systemów eksperckich opracowanych w języku CLIPS z programami opracowanymi w językach C lub C++. Zostanie przedstawiona procedura tworzenia plików z rozszerzeniem .exe możliwych do wywołania bezpośrednio ze środowiska DOS, procedura dołączania własnych aplikacji do innych programów oraz wiele informacji niezbędnych do przeprowadzenia procesów kompilacji i konsolidacji.

Ze względu na ograniczenia objętościowe w niniejszym opracowaniu zostaną przedstawione jedynie formy integracji programów w środowisku DOS. Procedurę uruchamiania programów w innych środowiskach, jak np. UNIX, można znaleźć w literaturze źródłowej, podanej w spisie literatury.

Kompilator języka CLIPS

W trakcie instalacji języka CLIPS, użytkownik ma możliwość dołączenia pełnej dokumentacji oraz zbiorów umożliwiających tworzenie wersji typu *.exe* dla własnych programów. Zbiory te są niezbędne do uruchomienia własnych systemów. Znajdują się one w katalogu SOURCE/CLIPS. Jest to pełny zestaw plików nagłówkowych oraz wzorców instrukcji dla języka C. Zbiory te, a jest ich kilkadziesiąt, należy skopiować do własnego katalogu razem z programem **setup.h** i **main.c** oraz własnymi programami, by zapewnić prawidłowe warunki kompilacji.

Opcje kompilatora są kontrolowane przez program **setup.h.** Program ten jest najważniejszy spośród plików nagłówkowych. Zawiera on globalne definicje używane przez CLIPS i wskaźniki dla kompilatora, które należy ustawić chcąc dołączyć go do środowiska. Dokładny opis wszystkich wskaźników jest umieszczony w programie **setup.h.** (i w literaturze). Zazwyczaj są one poprawnie ustawiane w trakcie instalacji i nie powinno się ich zmieniać w prostych implementacjach. Na przykład, podczas instalacji jest zwracana wartość bitu GENERIC z numerem systemu, na jaki program natrafił. Jest to wskaźnik określający typ użytego kompilatora-maszyny. Jeżeli chcemy go zmienić, musimy znać numer określający nasz system. Jeżeli wartość któregoś wskaźnika trzeba będzie zmienić, będzie to wyraźnie zaznaczone w tym tekście.

Chcąc traktować CLIPS jako narzędzie, program **setup.h.** należy umieścić w deklaracjach programu **main.c.** Jest to niezbędna informacja dla kompilatora o miejscu, w którym znajdują się deklaracje funkcji użytych przez system.

Dołączanie własnych aplikacji

W kolejnych rozdziałach zostaną przedstawione przykłady współpracy języka CLIPS z programami opracowanymi w języku C. Mówiąc o współpracy należy myśleć o czterech podstawowych formach wykorzystania języka CLIPS:

- wykorzystaniu funkcji opracowanych w języku C do efektywniejszej pracy systemu opracowanego w języku CLIPS;
- fragmentów systemu definiowanych w języku CLIPS;
- przesyłaniu danych między funkcjami opracowanymi w różnych językach;
- realizacji systemu eksperckiego w postaci wielu modułów dołączonych do programu głównego. Każdy z nich ma własny identyfikator i może być wywoływany w dowolnym miejscu programu głównego. Dzięki temu do szkieletu opracowanego w języku C można dołączać różne moduły, zmieniając czy też rozszerzając zastosowanie aplikacji.

Definiowane przez użytkownika funkcje

Jedną z istotnych zalet języka CLIPS jest możliwość dołączenia zewnętrznych funkcji oraz przesyłania argumentów między nimi. Funkcje mogą być opracowane w dowolnym środowisku mającym właściwości opisane w poniższym przykładzie:

```
int Define Function
                (functionName, functionType, functionPointer,
                actualFunctionName, functionRestriction);
char            *functionName, functionType, *actualFunctionName;
int             (*functionPointer)();
char            functionRestriction;
```

Korzystając z tych deklaracji można zdefiniować własną funkcję:

```
UserFunction()
{
        /* deklaracje własnych funkcji */
        extern double rta();
        extern VOID *dummy();
        DefineFunction ("rta",'d',PTIF rta,"rta","22n");
}
```

W tym przypadku pierwszy argument **rta** oznacza nazwę funkcji. Jest ona zgodna z nazwą użytą w opracowaniu własnych fragmentów programu w języku C. Drugi argument oznacza typ zmiennej zwracanej do środowiska CLIPS. Argumentem tym jest litera wybrana z listy przyporządkowującej jej określone znaczenie, np. literze *a* przyporządkowany jest adres, literze *d* zmienna w podwójnej precyzji, literze *f* zmienna jednobajtowa, literze *i* integer, literze *j* nieznana z opisu zmienna (np. symbol), literze *l* long integer, literze *n* integer, Float lub Instance Name itd.

Trzeci argument jest wskaźnikiem (pointerem) pokazującym adres kompilowanej funkcji.
Czwarty argument musi być identyczny z trzecim i jest łańcuchem (stringiem) reprezentującym funkcje opisane w języku C. Jest to nazwa używana następnie do wywołania funkcji z poziomu języka CLIPS.

W przypadku, gdy DefineFunction zwraca wartość różną od zera, deklaracja funkcji jest prawidłowa. W deklaracji funkcji może pojawić się piąty argument, który na ogół występuje w formie deklarowanej jako string. Wtedy jego znaczenie jest następujace:

<min-arg> <max-arg> [<default-type> <types>*]

Pierwsze dwie wartości muszą wystąpić w specyfikacji. Muszą one być cyframi lub literami. Cyfra odpowiadająca wartości <min-arg> określa najmniejszą liczbę argumentów wywoływanej funkcji. Jeżeli zostanie użyta na tej pozycji*, to oznacza, że argument ten nie jest określony. Analogicznie, druga cyfra określa maksymalną liczbę wywołań. Trzeci, niekonieczny argument określa typ wartości użytych w specyfikacji. Jako przykład można podać następujące wartości pola restriction:
"3*" minimalna wymagana liczba argumentów wynosi 3,
"*5" nieprzekraczalna liczba argumentów wynosi 5,
"35n" wymagana liczba argumentów wynosi od 3 do 5 i wszystkie powinny być typu integer lub float,
"66" wymagana liczba argumentów wynosi dokładnie 6.

Za pomocą tak zdefiniowanych funkcji można przesyłać informacje między deklarowanymi funkcjami a funkcjami języka CLIPS. W przykładzie użyto nazwy **rta** do deklaracji funkcji. Postać tej funkcji może być następująca:

```
#include "clips.h"
double rta()
{
        double podst, wys;
        if (ArgCountCheck("rta",EXACTLY,2) = = -1) return(-1.0);
        podst = RtnDouble(1);
        wys = RtnDouble(2);
        return(0.5 * podst * wys);
        /* pole trójkąta */
}
```

W tym momencie została zdefiniowana funkcja **rta**, której zadaniem jest obliczenie pola trójkąta o zadanych zewnętrznie parametrach. Przykład jej wykorzystania w środowisku języka CLIPS może być następujący:

```
CLIPS> (assert (pole_trojkąta (rta 5.0 10.0)))
CLIPS> (facts)
f-0 (pole_trojkąta 25.0)
For a total of 1 fact.
```

W wyniku wykonania funkcji ***assert*** na stosie faktów zostanie umieszczony nowy fakt z obliczonym przez zewnętrzną funkcję **rta** polem trójkąta. Funkcji o nazwie **rta** można używać na równych prawach z innymi funkcjami, zgodnie z gramatyką języka opisaną w pierwszym opracowaniu niniejszego cyklu. Aby stworzyć zbiór typu *.exe* należy skopiować wszystkie kody źródłowe do jednego katalogu, dołączyć wszystkie definicje zgodnie z podanymi przykładami oraz kod źródłowy systemu eksperckiego opracowanego w języku CLIPS, następnie poddać program **main.c** procesom kompilacji i linkowania.

Wywołanie systemów opracowanych w języku CLIPS w środowisku języka C

Każdy fragment systemu eksperckiego opracowanego w języku CLIPS może być wywołany w środowisku języka C. Wywołanie aplikacji CLIPS będzie działało na zasadzie wywołania procedury. Oczywiście w programie **main.c** muszą znaleźć się wszystkie wymagane deklaracje:

```
#include <stdio.h>
#include „clips.h"
```

W powyższych zbiorach znajdują się bowiem prototypy funkcji używanych w opracowaniu systemu eksperckiego. Oprócz tego program **main.c** musi zawierać instrukcję inicjującą **InitializeCLIPS**. Przykład programu **main.c** z deklaracjami zewnętrznej funkcji opracowanej w środowisku języka CLIPS może być następujący:

```
#include "clips.h"
main()
 {
 InitializeCLIPS();
 Load("constructs.clp");
 Reset();
 Run(-1L)
 }
UserFunctions()
 {
 extern VOID NazwaFunkcji();

 DefineFunction2("naz",'u',PTIF NazwaFunkcji, "NazwaFunkcji",
          "11n");
 }
```

Znaczenie poszczególnych elementów deklaracji zostało opisane w poprzednim rozdziale i nie zmieniło się ponieważ idea deklaracji jest tu bardzo podobna. W systemie opracowanym w języku CLIPS o nazwie **constructs.clp** (lub innym zgodnie z deklaracją) powinna znaleźć się definiowana funkcja **NazwaFunkcji**, w przykładzie użyto nazwy **naz**. Cały program tym razem jest uruchamiany w środowisku języka C i wykorzystuje procedurę **constructs.clp** opracowaną w języku CLIPS, której przykład jest prezentowany poniżej:

```
(deffacts init-data
  (data 34)
  (data 13.2))

(defrule get-data
  (data ?num)
  =>
  (printout t "Podaj nową wartość " ?num crlf)
  (assert (new-value (naz ?num))))

(defrule get-new-value
  (new-value ?num)
  =>
  (printout t crlf "Nowa wartość " ?num crlf))
```

Następnie należy skopiować wszystkie kody źródłowe programów i prototypów do jednego katalogu i poddać procesowi kompilacji i konsolidacji. W wyniku otrzymamy gotowy program w postaci przeznaczonej do bezpośredniego wywołania *.exe*.

Manipulacja obiektami

W tym punkcie zostanie przedstawiony przykład manipulowania obiektami. Podobnie, jak w poprzedniej części, rozpoczynamy od fragmentu kodu źródłowego napisanego w języku C, zawierającego odpowiednie deklaracje:

```
#include <stdio.h>
#include "clips.h"
main()
{
        VOID *c1, *c2, *c3;
        DATA OBJECT insdata,result;
        char numbuf[20];
        (initializeCLIPS();
        /* następnie należy załadować zdefiniowane obiekty */
        Load("complex.clp");
...}
```

W tym momencie obiekty są załadowane do środowiska języka C. Znajomość struktury obiektów pozwala na odpowiednie deklaracje zmiennych. Do wywołania systemu można wykorzystać polecenie **CLIPSFunctionCall** z argumentami określającymi nazwę handlera, nazwę klasy i typ rezultatu końcowego. Ze względu na rzadkość występowania tego typu aplikacji, zainteresowanych odsyła się do szczegółowych przykładów podanych w literaturze.

Kreowanie własnych systemów w środowisku języka C

W tej części zostanie przedstawiona procedura kreowania kompletnych systemów eksperckich opracowanych w języku CLIPS w środowisku języka C. Najistotniejszym elementem realizacji tego zadania jest kompilacja wszystkich struktur opracowanych w języku CLIPS do postaci jednego zbioru wykonywalnego z poziomu języka C. Do przekształcenia struktur systemu eksperckiego na kody źródłowe języka C należy użyć sekwencji poleceń:

```
CLIPS> (load "expert.clp")
CLIPS> (reset)
CLIPS> (construct-to-c exp 1)
```

Parametrami ostatniej instrukcji są:

<nazwa-zbioru> w naszym przypadku **exp**. Musi to być łańcuch lub symbol. Zaleca się użycie maksimum trójznakowej nazwy ze względu na modyfikacje wprowadzane w trakcie kompilacji. Wszystkie zbiory powstałe po wykonaniu powyższej instrukcji będą zaczynać się zaproponowaną nazwą.
<id> w naszym przypadku **1**. Jest to liczba określająca ten zbiór reguł jako cały moduł i będzie użyta w programie **main.c** jako identyfikator konkretnego systemu eksperckiego **exp**.
Opcjonalnie można użyć trzeciego elementu <max-elements> warunkującego wielkość zbiorów, które otrzymujemy w wyniku kompilacji. Kompilatory testowano na wielu maszynach, pod kątem możliwości uzyskania prawidłowych kodów źródłowych. Z doświadczeń wynika, iż wiele z nich wymaga użycia tego parametru. W szczególności w przypadku uruchamiania systemów eksperckich na IBM PC AT w środowisku języka Borland C++ sugeruje się użycie instrukcji (construct-to-c exp 1 300) z ograniczeniem wielkości zbiorów wyjściowych.

W wyniku użycia komendy (construct-to-c exp 1) otrzymujemy kilka zbiorów typu **exp1_1.c, exp1_2.c, ..., exp7_1.c.** Charakteryzują się one tym, iż mają identyczną część nazwy, zgodnie z deklaracją w pierwszym argumencie użytej funkcji. Ta część powinna być tak dobrana, aby pełna długość nazwy była akceptowana przez środowisko (np. MS-DOS interpretuje jedynie osiem pierwszych znaków nazwy).

Następną czynnością konieczną do uruchomienia systemu jest zmiana bitu RUN-TIME z zera na jedynkę. Należy to zrobić w programie **setup.h**: RUN-TIME = 1. Innych bitów nie należy zmieniać.

System ekspercki może być wywołany z poziomu języka C w następujący sposób:

```
#include <stdio.h>
#include "clips.h"
main()
{
        InitializeCLIPS();
        InitImage 1();
        Reset()
        Run(-1L);
}
```

W programie tym argumentem funkcji **InitImage** jest identyfikator określający konkretny moduł skompilowanego wcześniej systemu (w przykładzie użyto liczby 1). Ostatnim krokiem jest poddanie wszystkich zbiorów procesowi konsolidacji. W wyniku tego otrzymujemy program możliwy do wykonania.

Możliwości wykorzystania języka CLIPS w innych środowiskach

W dotychczasowych rozważaniach przedstawiono przykłady współpracy systemów eksperckich, napisanych w języku CLIPS, z programami opracowanymi w środowisku języka C.
Zainteresowanych odsyła się do literatury w celu poznania możliwości uruchomienia własnych aplikacji w innych środowiskach. Warto dodać, iż współpraca taka jest

możliwa po spełnieniu następujących warunków:
- istnieje możliwość opracowania programu głównego (main program) w użytym środowisku;
- są dostępne funkcje dla języka C z poziomu użytego środowiska;
- funkcje opracowane w innym środowisku mogą być wywoływane przez język CLIPS.

Wymienione trzy warunki daje się sprowadzić do generalnego wniosku: jeżeli jest możliwe połączenie aplikacji opracowanych w języku C z innym środowiskiem, to również jest możliwe dołączenie systemów eksperckich opracowanych w języku CLIPS. Na przykład, w literaturze są podawane przykłady dołączania systemów do takich programów jak, Ada czy FORTRAN. Kompilator języka CLIPS był testowany na takich maszynach, jak VAX (VMS), SUN (UNIX), Macintosh (C i MPW C), IBM PC AT (dla różnych kompilatorów m.in. Microsoft C V6.0 w środowisku MS-DOS, Borland C++ V3.1 w środowisku Windows 3.1 i MS-DOS). Na wszystkich maszynach uzyskano prawidłowe wyniki kompilacji systemu eksperckiego opracowanego w języku CLIPS na kod źródłowy w języku C przy użyciu funkcji **(construct--to-...)**.

* * *

W omawianych czterech artykułach, opublikowanych w kolejnych numerach INFORMATYKI został przedstawiony jedynie zarys możliwości języka CLIPS. Podstawowa literatura obejmuje kilkaset stron tekstu, pełna instalacja wymaga 12MB pamięci na dysku. Już tylko te dane pokazują potencjalne możliwości tego produktu. Ze względu na swoją specyfikę oraz liczne zalety jest on bezpłatnie dostępny w sieci INTERNET.

W kolejnych pracach cyklu szczególną uwagę zwrócono na:
- znaczenie poszczególnych funkcji oraz przykłady ich wykorzystania w większych strukturach typu *reguły* czy *handlery;*
- strategie i mechanizmy wnioskowania w systemach eksperckich;
- podstawy programowania obiektowego;
- zasady kompilacji gotowych aplikacji i wykorzystywania ich w innych środowiskach.

Łatwość opracowania pewnych, specyficznych zadań przy dużej skuteczności pracy może spowodować, iż to własnie język CLIPS zostanie wybrany przez programistów do realizacji systemów eksperckich i decyzyjnych. Znane są już działajace aplikacje zrealizowane w języku CLIPS, z powodzeniem realizujące zadania trudne do łatwego oprogramowania w innym środowisku. Autor niniejszego opracowania zachęca do wykonania pierwszego kroku w ciemno i użycia języka CLIPS. Jest przekonany, że w wielu przypadkach okaże się on słuszny. Chętnie poznałby wszelkie spostrzeżenia na temat tego cyklu opracowań, jak również utworzonych systemów eksperckich. Dlatego też na zakończenie podaje adres poczty elektronicznej, pod którym oczekuje na wszelkie listy i uwagi: **pytlak@ipe.pw.edu.pl**

LITERATURA

[1] CLIPS Reference Manual, Vol. I: Basic Programming Guide, CLIPS Version 6.0, Software Technology Branch, Lyndon B. Johnson Space Center, 1995

[2] Ignizio J.P.: Introduction to expert systems. McGraw-Hill Inc. New York, 1991.

ARIS Toolset...

dokończenie ze s. 13

[9] Scheer A.-W.: *Architektur Integrierter Informationssysteme,* Springer-Verlag 1991. (Wersja w jęz. angielskim: Scheer A.-W.: Architecture of Integrated Information Systems, Springer-Verlag 1992)

[10] Scheer A.W.: *Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse,* Springer-Verlag 1994. (Wersja w jęz. angielskim: Scheer A.-W.: Business Process Engineering. Reference Models for Industrial Enterprises, Springer-Verlag 1994)

[11] Scheer A.-W., Elsner T., Dziuba D., Lasek M.: ARIS – Architektura Zintegrowanych Systemów Informacyjnych, *Informatyka* nr 9, s. 8–12, 1994

[12] Schulte R., Rosemann M., Rotthowe Th.: Business Proess Reengineering in Theorie und Praxis, *m & c* 3/94, S. 211–219.

Zagadnienia pozyskiwania wiedzy...

dokończenie ze s. 17

Należy wobec tego wziąć pod uwagę maksymalną szybkość przemieszczania się układu w przestrzeni stanów (v_i) oraz odległość między maksimami funkcji przynależności do sąsiednich zbiorów rozmytych (x_i) w pobliżu punktu pracy (celu sterowania) dla każdej zmiennej układu sterowania. Maksymalny czas próbkowania (t_p) można wyznaczyć za pomocą wyrażenia:

$$\{4\} \qquad t_n \leq \min_i \left(\frac{x_i}{v_i}\right)$$

Należy jednak wziąć pod uwagę fakt, że podany wyżej wzór jest tylko wartością szacunkową i rzeczywisty okres próbkowania powinien być kilkakrotnie niższy.

Znalezienie reguł dla regulatora rozmytego tylko na podstawie wiedzy eksperta jest zagadnieniem dość złożonym i trudnym do przeprowadzenia, bez dobrej znajomości zarówano obiektu, jak i właściwości sterowników rozmytych. Decydującym sprawdzianem dla tak dobranego układu muszą być symulacje i poprawki wprowadzane na ich podstawie. Ręczne znajdowanie i dokonywanie tych poprawek jest dość żmudne i pracochłonne, dlatego wskazane jest wykorzystanie opracowanych do tego celu metod. Do znalezienia optymalnego zestawu reguł dla układu opartego na logice rozmytej często stosuje się techniki oparte na statystycznej analizie danych lub też wykorzystuje się do tego celu sieci neuronowe.

(cdn.)

LITERATURA

[1] Czogała E., Pedrycz W.: Elementy i metody teorii zbiorów rozmytych. Warszaw, PWN 1985.

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Francusko-Polska Wyższa Szkoła Nowych Technik Informatyczno-Komunikacyjnych w Poznaniu), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), dr inż. Krzysztof SACHA (Politechnika Warszawska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Zbigniew ZIELIŃSKI (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) – przewodniczący zespołu.

ROK III ● LUTY ● 1996 ● 15

Metody analizy i specyfikacji wymagań

Strukturalno-graficzny język opisu i projektowania aplikacji czasu rzeczywistego

Tomasz Szmuc, Piotr Szwed
Katedra Automatyki
Akademia Górniczo-Hutnicza
Kraków

W artykule przedstawiono język graficznej specyfikacji aplikacji czasu rzeczywistego LACATRE (*Language d'Aide a la Conception d'Aplications Temps-REel*). Język ten umożliwia dekompozycję aplikacji i hierarchiczne definiowanie jej funkcji i struktury zgodnie z metodyką SART [10]. Z drugiej strony obiekty języka modelują mechanizmy dostarczane z poziomu systemu operacyjnego, co stwarza potencjalne możliwości definiowania wykonalnych specyfikacji. W artykule przedstawiono podstawowe elementy języka oraz pokazano prosty przykład jego zastosowania przy projektowaniu aplikacji czasu rzeczywistego.

Początkowe fazy cyklu wytwarzania oprogramowania mają duży wpływ na efekty końcowe, a niekiedy decydują wręcz o powodzeniu całego przedsięwzięcia. Ponad 70% błędów popełnianych przy tworzeniu systemów ma swoje źródło w fazach specyfikacji i projektowania [2]. Naturalne jest zatem dążenie do rozwijania narzędzi wspomagających tworzenie oprogramowania w tych fazach. W artykule prezentowany będzie język LACATRE (LA4) [3, 4, 9], który może być stosowany do graficznej specyfikacji oprogramowania czasu rzeczywistego [8]. Obiekty tego języka modelują mechanizmy dostarczane z poziomu systemu operacyjnego, co umożliwia weryfikację strukturalnej poprawności [7] tworzonej aplikacji, jak również semi-automatyczną generację kodu. Natomiast hierarchiczne rozszerzenia LA4 dostarczają narzędzi do strukturalnego opisu specyfikacji i projektu. W artykule główny nacisk położono na stronę prezentacji języka i jego zastosowania w fazach wstępnego projektowania oprogramowania. Zagadnienia analizy i weryfikacji nie są tu przedstawiane, ze względu na ograniczenia artykułu. Odpowiednie informacje można znaleźć w pracach [7, 8].

Podstawowym celem projektu LACATRE było zdefiniowanie języka wspomagającego specyfikowanie i projektowanie aplikacji systemu iRMX. Zakładano, że we wstępnych fazach projektu należy skupić się na przeprowadzeniu strukturalnej dekompozycji aplikacji na zadania i doborze mechanizmów komunikacji i synchronizacji, pozostawiając szczegóły realizacyjne, uzależnione od docelowego języka programowania do dalszych etapów [3].

Język LACATRE (francuski skrót: LA4) jest przeznaczony do opisywania wielozadaniowych aplikacji czasu rzeczywistego w początkowych fazach cyklu życia oprogramowania [1]. Jego zastosowanie ułatwia przejście od strukturalnego modelu analizy (agencje i procesy) do wykonalnego modelu projektu wstępnego (obiekty i funkcje pierwotne), i dotarcie do projektu wstępnego szczegółowego (formy algorytmiczne). Zarówno język, jak i związana z nim metodyka projektowania pozwalają przeprowadzić dekompozycję aplikacji na odrębne zadania uwzględniając występujące w systemach czasu rzeczywistego mechanizmy komunikacji i synchronizacji. Język umożliwia stosowanie strukturalnego podejścia do opracowania aplikacji, zgodnego z metodyką SART [10], nie pozwalając jednakże w obenej wersji na pełne modelowanie przepływu danych między składowymi funkcjonalnymi aplikacji.

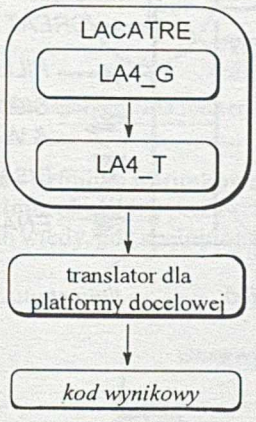
LA4 może być używany jako język specyfikacji oprogramowania dla takich systemów czasu rzeczywistego, jak: iRMX, iRMK, VRTX 32, VxWorks, OS-9 i CRAFT. Język ten definiuje większość obiektów i funkcji występujących w tych systemach; najbliższy jest jednak środowisku iRMX.

Język LA4

Język LA4 jest zdefiniowany w dwóch postaciach: graficznej (LA4_G) i tekstowej (LA4_T). Specyfikacja graficzna jest dwuwymiarowym diagramem obrazującym w precyzyjny i syntetyczny sposób budowę wielozadaniowej aplikacji czasu rzeczywistego. Schemat zawiera graficzną reprezentację wszystkich statycznych składowych aplikacji, nazywanych *obiektami*, oraz mechanizmów synchronizacji i komunikacji, realizowanych przez wywołania *funkcji pierwotnych*. Obiekt jest przedstawiany na schemacie w postaci symbolu graficznego uzupełnionego o listę parametrów; wywołania funkcji są zaznaczane jako połączenia między obiektami (zwykle linie łamane z dołączonym symbolem graficznym pozwalającym rozróżnić typ funkcji). Powyższe elementy są uzupełnione przez *formy algorytmiczne* odpowiadające strukturalnym instrukcjom języków programowania wysokiego poziomu. Gramatyka języka LA4 definiuje ściśle symbolikę używaną do reprezentacji obiektów i funkcji, a także reguły połączeń.

Kod LA4_T jest równoważny syntaktycznie diagramowi LA4_G i jest zazwyczaj generowany na podstawie specyfikacji graficznej LA4_G.

Budowa aplikacji za pomocą systemu LA4 składa się z dwóch etapów. W pierwszym określa się strukturę aplikacji, używając w tym celu narzędzia graficznego; otrzymany schemat wzajemnych powiązań obiektów (LA4_G) jest automatycznie tłumaczony do ekwiwalentnej postaci kodu (LA4_T). W drugim, zależnie od wybranego docelowego systemu operacyjnego i języka programowania, jest uruchamiany odpowiedni moduł translatora generującego szkielet programu czasu rzeczywistego (rys. 1).



Rys. 1. Cykl budowy aplikacji czasu rzeczywistego

Należy zauważyć, że nie zawsze istnieje pełna odpowiedniość pomiędzy obiektami oraz funkcjami pierwotnymi LA4, a elementami wybranego systemu operacyjnego. Brak pełnej standaryzacji w tej dziedzinie uniemożliwia – jak dotąd – stworzenie uniwersalnego języka specyfikacji.

Podobieństwo zasadniczych elementów i mechanizmów jądra systemów operacyjnych czasu rzeczywistego pozwoliło jednak na wyodrębnienie nadzbioru obejmującego najczęściej używane typy obiektów i funkcji oraz zdefiniowanie go w gramatyce języka LA4. Wymienić tu należy:

- **Wielozadaniowość.** Wszystkie funkcje aplikacji czasu rzeczywistego dzielone są między współbieżne procesy. Pozwala to na łatwiejsze dostosowanie funkcji aplikacji do asynchronicznych zdarzeń napływających z zewnątrz. Zadanie (ang. *task*) jest podstawowym obiektem programowalnym LA4.
- **Priorytetowy mechanizm szeregowania zadań z wywłaszczaniem.** Mechanizm ten pozwala na wykonywanie wielu procesów (zadań) w kolejności uzależnionej od ich ważności. Znaczenie nadane zewnętrznym zdarzeniom odzwierciedla się poprzez kolejność przydziału procesora do procedur obsługi. Zadanie o najwyższym priorytecie, które jest *gotowe* (nie jest w stanie oczekiwania na komunikację lub zasób) wywłaszcza wykonywane zadanie o niższym priorytecie i otrzymuje przydział procesora. W języku LA4 uwzględniono grupę funkcji pierwotnych pozwalających na ingerencję w mechanizmy szeregowania zadań.
- **Mechanizmy komunikacji i synchronizacji.** W systemie składającym się z wielu zadań muszą istnieć mechanizmy wymiany informacji i synchronizacji. Należą do nich zazwyczaj semafory i kolejki komunikatów.
- **Mechanizmy ochrony i podziału zasobów systemowych.** Obiekt *resource* języka LA4 i grupa związanych z nim funkcji pierwotnych pozwalają na specyfikację mechanizmów ochrony i dostępu do zasobów.
- **Wymiana informacji między zadaniami i przerwaniami.** Informacje o zewnętrznych zdarzeniach pojawiają się zazwyczaj asynchronicznie w postaci przerwań. Komunikacja między procedurami obsługi przerwań i zadaniami pozwala na przesunięcie wszystkich długotrwałych operacji obsługi (buforowanie, uwzględnianie priorytetów) do zadań, tak aby zminimalizować czas powrotu systemu do stanu gotowości na przyjęcie zdarzenia zewnętrznego. W języku LA4 zdefiniowano funkcje pierwotne SIGNAL_IT (WAIT_IT) pozwalające na komunikację pomiędzy zadaniami i przerwaniami. Dla wielu systemów operacyjnych komunikacja może odbywać się także za pośrednictwem semaforów i kolejek komunikatów.
- **Możliwość zaprogramowania zadań uruchamianych przez system.** Większość systemów czasu rzeczywistego pozwala na okresowe wywoływanie wybranych procedur lub w odpowiedzi na pojawienie się sytuacji wyjątkowej. Rolę tę spełnia w języku LA4 obiekt *alarm*.

Elementy języka

Obiekty LA4. Rozróżnia się dwa typy obiektów: *programowalne* i *konfigurowalne*. Do pierwszej grupy zalicza się: zadanie (ang. *task*), procedurę obsługi przerwania (ang. *interrupt*) i procedurę uśpioną (ang. *alarm*). Obiekty programowalne odzwierciedlone są w kodzie LA4_T

jako procedury zawierające ciągi instrukcji odpowiadających wołanym funkcjom pierwotnym. Obiekty konfigurowalne: semafor (ang. *semaphore*), kolejka komunikatów (ang. *mailbox*), komunikat (ang. *message*), tablica komunikatów (ang. *message pool*), zasób (ang. *resource*) i zdarzenie (ang. *event*), jawnie nie wykonują żadnych akcji, dlatego też w kodzie LA4_T pojawiają się jedynie jako argumenty wywołań funkcji pierwotnych.

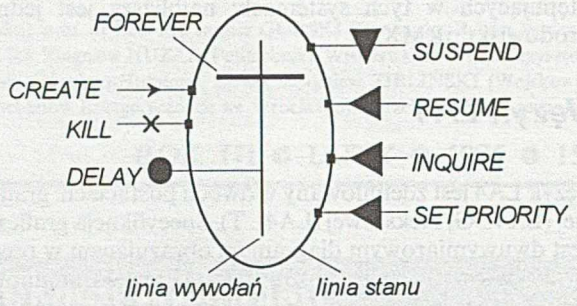
Graficzne symbole obiektów językaLA4_G składają się z kilku rodzajów linii. Zależnie od typu, obiekty mogą zawierać linie stanu (ang. *state bars*), linie wywołań (ang. *progress bars*) i ewentualnie linie akcji (ang. *action bars*). *Linia stanu* jest zdefiniowana dla każdego obiektu; do niej dochodzą linie łamane reprezentujące funkcje pierwotne kreacji, destrukcji i zmiany stanu. *Linia wywołań* jest zdefiniowana jedynie dla obiektów programowalnych. Od tej linii odchodzą linie reprezentujące funkcje pierwotne wywoływane przez obiekt programowalny. Kolejność punktów zaczepienia na linii wywołań jest zgodna z kolejnością wykonywania instrukcji. Linie wywołań obiektów programowalnych mogą zostać rozszerzone i rozgałęzione przez użycie form algorytmicznych *IF...THEN...ELSE, WHILE, REPET...UNTIL, SWITCH...CASE, FOREVER* (pętla nieskończona) oraz *PROCEDURE*. Symbole form algorytmicznych określają alternatywne drogi przepływu sterowania. *Linie akcji* występują w obiektach konfigurowalnych. Linie te łączą się z końcami linii łamanych funkcji pierwotnych specyficznych dla danego obiektu programowalnego.

Funkcje pierwotne LA4. Funkcje pierwotne LA4 zapewniają jednolity, niezależny od systemu operacyjnego interfejs do funkcji systemowych. Odpowiednikiem funkcji LA4 jest wywołanie jednej lub kilku funkcji jądra systemu operacyjnego. Wyróżnia się trzy zasadnicze grupy funkcji pierwotnych:

- **Funkcje stanu.** Pozwalają na zmianę lub odczytanie stanu obiektu. Należą do nich globalne funkcje kreacji i destrukcji oraz funkcje typu aktywacji, uśpienia zadań, maskowania przerwań. W gramatyce LA4_G graficzne symbole funkcji stanu dochodzą zazwyczaj do linii stanu obiektów.
- **Funkcje akcji.** Umożliwiają ustanowienie połączenia (synchronizacji lub komunikacji) między dwoma obiektami programowalnymi za pośrednictwem obiektu programowalnego (semafor, kolejka komunikatów). Funkcje te są zazwyczaj specyficzne dla obiektu konfigurowalnego; ich symbole graficzne dochodzą do linii akcji.
- **Funkcje synchronizacji.** Grupa ta obejmuje funkcje bezpośrednio synchronizujące dwa obiekty programowalne (bez pośrednictwa innego obiektu programowalnego). W kodzie LA4_T pojawiają się jako komplementarne pary instrukcji w ciele obu uczestniczących w komunikacji obiektów programowalnych.

Zestawienie elementów języka

Zadanie (ang. *task*) jest niezależnym procesem aplikacji, współistniejącym z innymi procesami. Zadania w systemie mają swój własny kontekst pamięciowy (stos, rejestry) przydzielany im w momencie inicjalizacji (funkcja pierwotna CREATE) i zwalniany w momencie usuwania z systemu (KILL). Typowe stany zadania to: gotowość do przejęcia sterowania (ang. *ready*), stan wykonywania (ang. *running*), oczekiwania na zasób (ang. *waiting*), uśpienia (ang. *delayed*) lub zawieszenia (ang. *pending*). Graficzny symbol zadania jest pokazany na rys. 2.

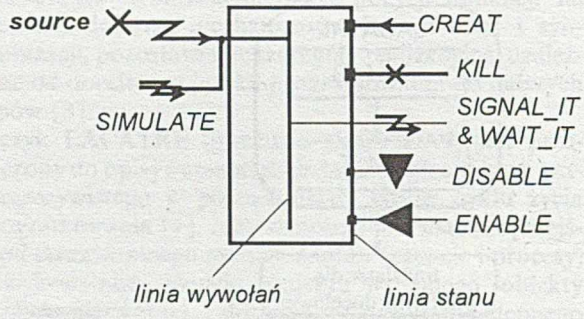


Rys. 2. Graficzny symbol zadania

Zazwyczaj zadania wykonują w pętli nieskończonej (FOREVER) powtarzalne ciągi instrukcji poprzedzone sekwencją inicjalizacyjną. Funkcje pierwotne LA4 związane bezpośrednio z zadaniami: DELAY (uśpienie zadania na dany okres), SUSPEND i RESUME (zawieszenie i uaktywnienie zadania), INQUIRE (odczyt stanu), SET_PRIORITY (zmiana priorytetu).

Zadanie jest obiektem programowalnym, stąd w reprezentacji LA4_T pojawia się w postaci ciągu instrukcji ograniczonych słowami kluczowymi TASK...TASK_END.

Przerwanie (ang. *interrupt*) reprezentuje procedurę obsługi przerwania. Systemy operacyjne czasu rzeczywistego dostarczają zazwyczaj funkcje umożliwiające dołączenie napisanej przez programistę procedury obsługi do wektora przerwania. Procedury tego typu poddane są rozmaitym ograniczeniom, przykładowo: jest niedopuszczalne umieszczanie wywołań funkcji jądra, które mają nieokreślony czas wykonania (np.: czekanie na semafor) mogących zablokować system. Prawidłowe zarządzanie wszystkimi elementami (tablica wektorów przerwań, procedura obsługi, sygnalizacja przerwań) musi być zazwyczaj zrealizowane przez programistę aplikacji. Symbol obiektu przerwania jest pokazany na rys. 3.

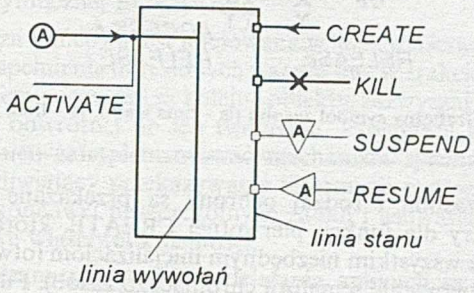


Rys. 3. Graficzny symbol przerwania

Funkcja CREATE dołącza procedurę obsługi przerwania do wektora; KILL przywraca stan początkowy. Natomiast DISABLE i ENABLE odpowiednio blokują i odblokowują obsługę przerwań; SIMULATE umożliwia programowe uruchomienie procedury obsługi przerwania.

W ciele tego obiektu jest możliwe umieszczanie wywołań dopuszczalnych przez system funkcji pierwotnych (niedozwolone użycie wykrywane jest przez translator). Język definiuje także dwie funkcje bezpośredniej synchronizacji: SIGNAL_IT – wykonywana wewnątrz procedury i WAIT_IT wywołana przez zadanie.

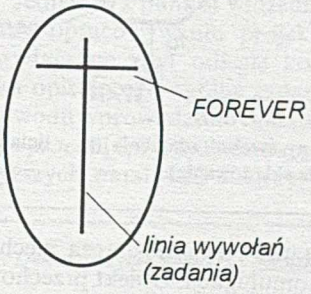
Alarm (ang. *alarm*) reprezentuje procedurę wywoływaną cyklicznie, z opóźnieniem lub w wyniku reakcji na sytuację wyjątkową. Jest ona zazwyczaj poddana podobnym ograniczeniom, jak procedury obsługi przerwań, ponieważ podobnie jak one, nie posiada własnego kontekstu pamięciowego (jest wykonywana bez przeładowania). Symbol alarmu jest przedstawiony na rys. 4.



Rys. 4. Graficzny symbol alarmu

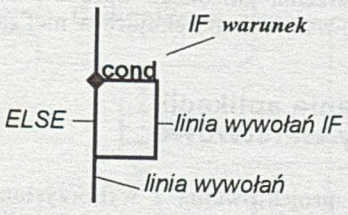
Formy algorytmiczne stanowią uzupełnienie konstrukcji języka LA4 o instrukcje strukturalne. W gramatyce języka są one traktowane jako uzupełnienia lub rozszerzenia linii wywołań obiektów programowalnych. Podstawowe formy algorytmiczne:

- **forever** – początkuje ciąg instrukcji wykonywany cyklicznie; instrukcje umieszczone na linii wywołań przed początkiem pętli *forever* wykonywane są jednorazowo po inicjalizacji zadania (rys. 5);



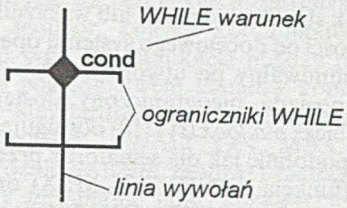
Rys. 5. Forma algorytmiczna forever

- **if-then** – definiuje warunkowe rozgałęzienie obliczenia algorytmu, określone przez sekwencję funkcji wykonywanych wtedy, gdy spełniony jest zadany warunek (rys. 6);



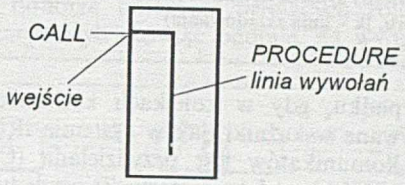
Rys. 6. Forma algorytmiczna if-then

- **while** – definiuje powtarzalny fragment obliczenia, określony przez sekwencję funkcji wykonywanych tak długo, jak długo jest spełniony zadany warunek (rys. 7);



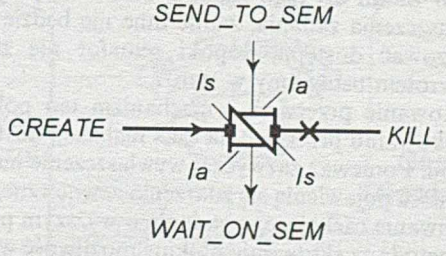
Rys. 7. Forma algorytmiczna while

- **repeat-until** – definiuje powtarzalny fragment obliczenia, określony przez sekwencję funkcji wykonywanych do momentu spełnienia zadanego warunku [9];
- **switch-case** – definiuje warunkowe rozgałęzienie obliczenia, w którym sekwencja wykonanych akcji zależy od wartości zmiennej sterującej [9];
- **procedure** – reprezentuje procedurę (kod) dzieloną przez zadania. Procedura nie jest obiektem czasu rzeczywistego, ponieważ jej wykonanie odbywa się w kontekście pamięci wołającego zadania (rys. 8).



Rys. 8. Forma algorytmiczna procedure

Semafor (ang. *semaphore*) jest klasycznym mechanizmem systemów czasu rzeczywistego używanym do ochrony zasobów lub synchronizacji. W języku LA4 obiekt ten jest semaforem całkowitym: posiada licznik i kolejkę, w której przechowywane są zlecenia obiektów czekających na jednostki. Funkcja SEND_TO_SEM zwiększa licznik semafora o daną liczbę jednostek; funkcja WAIT_ON_SEM powoduje umieszczenie zlecenia w kolejce i przejście wołającego zadania w stan oczekiwania. Symbol semafora jest pokazany na rys. 9.

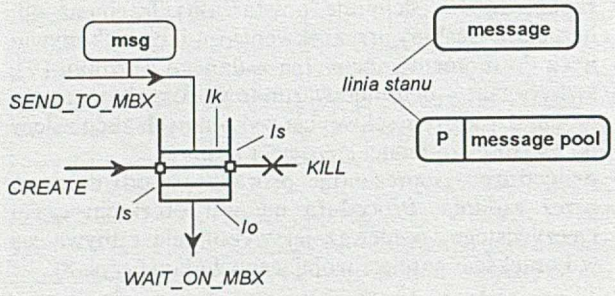


Rys. 9. Graficzny symbol semafora (ls – linia stanu, la – linia akcji)

Kolejka zleceń może być zarządzana według strategii FIFO (zlecenia obsługiwane są w kolejności składania) lub według priorytetu zadań. Zadanie przechodzi w stan gotowości, jeśli w semaforze złożona zostanie liczba

jednostek wystarczająca do realizacji zlecenia – licznik jednostek semafora jest wówczas odpowiednio zmniejszany.

Kolejka komunikatów pośredniczy w asynchronicznej komunikacji między zadaniami. Komunikat jest tu traktowany jak struktura danych lub wskaźnik do struktury (w zależności od docelowego systemu operacyjnego). Obiekt programowalny po utworzeniu lub otrzymaniu komunikatu posiada nieograniczony dostęp do jego zawartości. Obiekt *mailbox* (rys. 10) obsługuje w zasadzie dwie kolejki (podobnie jak dla semafora): przechowującą komunikaty (funkcja SEND_TO_MBX) oraz gromadzącą zlecenia (funkcja WAIT_ON_MBX).

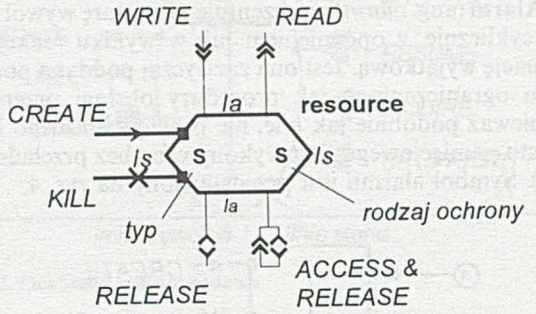


Rys. 10. Graficzny symbol kolejki komunikatów (ls – linia stanu, lo – linia odbioru, lk – linia składowania)

W przypadku, gdy w kolejkach komunikatów są przechowywane wskaźniki (jak w systemie iRMX), pamięć dla komunikatów jest przydzielana (CREATE) i zwalniana (KILL) z dzielonej sterty. Ze względu na to, że działanie funkcji realizujących przydział pamięci jest niedeterministyczne i może nieść znaczne opóźnienia czasowe uwzględniono dodatkowy obiekt często spotykany w tego typu systemach – reprezentujący jednorazowo inicjowaną tablicę komunikatów (ang. *message pool*).

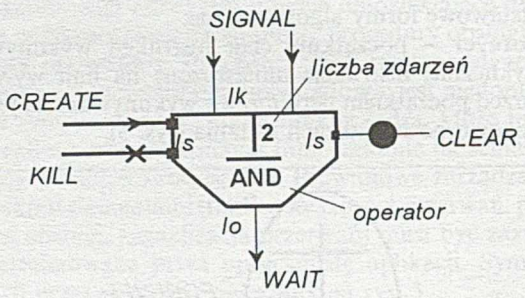
Zasób (ang. *resource*) reprezentuje krytyczny zasób systemu: wspólną pamięć, lokalne lub zdalne pliki oraz urządzenia. W języku LA4 uwzględniono podstawowe mechanizmy ochrony zasobów:

- Semafor binarny – dostęp do zasobu chroniony jest przez semafor. Semafor jest wstępnie ustawiony na *1*. Przed rozpoczęciem właściwego dostępu do zasobu zadanie ustawia semafor w stan *0* i blokuje dostęp innych zadań do tego zasobu. Nawet w przypadku wywłaszczenia zadania, żadne inne nie będzie mogło zrealizować dostępu, dopóki semafor nie zostanie z powrotem ustawiony w stan *1*.
- Maskowanie przerwań – mechanizm ten polega na zablokowaniu przerwań na czas realizacji dostępu do zasobu. Ponieważ zazwyczaj wywłaszczenie następuje w wyniku pojawienia się zdarzenia zewnętrznego i odblokowania czekającego zadania o wyższym priorytecie, metoda ta skutecznie blokuje możliwość wywłaszczania. Nie jest jednak zalecana ze względu na niezdolność systemu do reagowania na zdarzenia zewnętrzne w trakcie realizacji dostępu do zasobu przez jedno z zadań.
- Blokada zadań – mechanizm ten polega na blokadzie systemu szeregowania na czas realizacji dostępu do zasobu. Możliwa jest jednak rejestracja napływających zdarzeń zewnętrznych (przerwań). Mechanizm ten nie gwarantuje jednak, że przerwania będą obsłużone w określonym czasie, co jest zazwyczaj wymaganiem stawianym aplikacjom czasu rzeczywistego.



Rys. 11. Graficzny symbol zasobu (ls – linia stanu, la – linia akcji)

Typ zasobu i rodzaj ochrony są przekazane jako parametry dla funkcji pierwotnej CREATE, która odpowiada wszystkim niezbędnym inicjalizacjom (otwarcie pliku, utworzenie semafora chroniącego zasób). Funkcje ACCESS, RELEASE są standardowymi akcjami pobrania i zwolnienia zasobu, natomiast KILL przeprowadza czynności końcowe (zamknięcie pliku). Symbol zasobu jest przedstawiony na rys. 11.



Rys. 12. Graficzny symbol zdarzenia (ls – linia stanu, lo – linia odbioru, lk – linia składowania)

Zdarzenie (ang. *event*) dostarcza mechanizmów synchronizacji i komunikacji. Obiekt przechowuje flagi zdarzeń elementarnych i ustawia odpowiednio flagę zdarzenia złożonego, jeśli flagi zdarzeń elementarnych są połączone operatorem AND lub OR. Linia SIGNAL_EVENT sygnalizuje zajście zdarzenia elementarnego; WAIT_EVENT zawiesza działanie wołającego zadania aż do momentu pojawienia się zdarzenia złożonego. CLEAR_EVENT zeruje flagi zdarzeń. W przeciwieństwie do kolejek zleceń dla obiektów *semaphore* i *mailbox*, pojawienie się zdarzenia złożonego opróżnia całą kolejkę odblokowując wszystkie oczekujące w niej zadania.

Fazy działania aplikacji i schematy warstwowe

W metodyce projektowania z wykorzystaniem języka LA4 wyróżnia się trzy fazy działania aplikacji czasu rzeczywistego: wstępną, właściwą i końcową.

W trakcie fazy wstępnej są tworzone wszystkie elementy składowe aplikacji, niezbędne do przejścia do fazy właściwej. Wywoływane są tu globalne funkcje pierwotne kreacji, synchronizowane, jeśli zachodzi taka potrzeba, przez odpowiednie semafory. Akcje związane z fazą wstępną są wykonywane przez zadania tylko raz; z tego powodu wszelkie inicjalizacje są umieszczane zazwyczaj na linii ich wywołań, przed formą algorytmiczną FOREVER.

W fazie właściwej funkcje są realizowane zgodnie z przeznaczeniem aplikacji. Faza ta powinna zostać wyraźnie oddzielona od wstępnej. W typowych zastosowaniach w fazie właściwej zadania wykonują powtarzające się akcje zdefiniowane na linii wywołań poniżej formy algorytmicznej FOREVER.

Faza końcowa jest inicjowana w fazie właściwej w wyniku spełnienia określonych warunków. W trakcie tej fazy z systemu usuwane są kolejne obiekty, zazwyczaj w kolejności odwrotnej do ich tworzenia. Projektant aplikacji powinien zaimplementować mechanizm synchronizacji umożliwiający przekazywanie informacji o przejściu do fazy końcowej poszczególnych zadań i usuwania obiektów we właściwej kolejności.

Opisane powyżej fazy działania aplikacji są w pełni separowalne. Symbolika graficzna LA4 dopuszcza odzwierciedlenie działań podjętych w poszczególnych fazach jako warstw schematu aplikacji. W ten sposób zwiększa się przejrzystość dokumentacji graficznej i zapewnia się metodyczne podejście do projektu.

Hierarchiczne rozszerzenia języka LA4

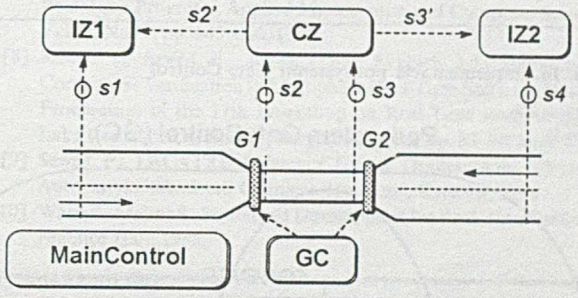
Podstawowe elementy języka LA4: obiekty, formy algorytmiczne i funkcje pierwotne pozwalają modelować mechanizmy oferowane przez systemy operacyjne czasu rzeczywistego. Jednakże z punktu widzenia programisty, szczególnie przy opracowywaniu projektu wstępnego, elementy te wydają się zbyt odległe koncepcyjnie od poziomu języka opisującego ogólne zadania aplikacji.

Z tego też powodu wprowadzono dodatkowe elementy języka: *agencję* i *proces* [6, 7], umożliwiające specyfikację systemu na wyższych warstwach abstrakcji.

Agencja jest traktowana jako spójny zbiór aktywnych i nieaktywnych elementów realizujących określoną funkcję wewnątrz programu. Agencja składa się z procesów, czyli aktywnych elementów nie mogących jednakże istnieć w pełni samodzielnie oraz kanałów komunikacyjnych, których rolę pełnią klasyczne obiekty konfigurowalne, jak semafor, kolejka komunikatów czy zasób określonego typu. Komunikacja między procesami odbywa się za pośrednictwem kanałów komunikacyjnych z użyciem standardowych funkcji pierwotnych, jak SEND_TO_SEM, WAIT_ON_MBX. Definiując agencję wyróżnia się także, zgodnie z metodologią SART, jej elementy terminalne (ang. *borders*). Mogą one reprezentować obiekty LA4 pełniące rolę interfejsu do obiektów fizycznych (czujnik, terminal), kanały komunikacji przy współpracy z innymi agencjami aplikacji, lub też punkty startowe procedur inicjalizacji i destrukcji.

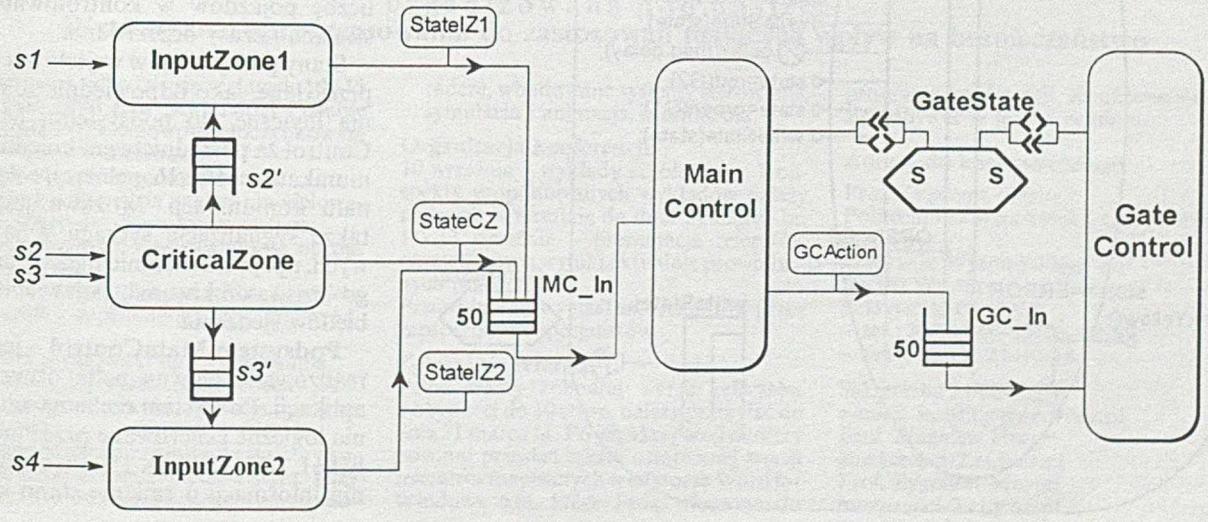
Proces reprezentuje aktywną funkcję składową aplikacji. Może być hierarchicznie rozłożony na dalsze podprocesy. Z punktu widzenia projektowania jest najważniejsze wydzielenie jego interfejsu, tzn. podanie zbioru niezbędnych kanałów komunikacyjnych i obsługujących je funkcji pierwotnych. Sama implementacja procesu może być zrealizowana na najniższym szczeblu hierarchii z użyciem języka LA4 lub z wykorzystaniem języka programowania docelowej platformy.

Cykl projektowania aplikacji polega więc w fazie wstępnej na podziale jej funkcji między agencje i podprocesy, doborze obiektów pełniących rolę kanałów komunikacji, czyli mówiąc ogólnie, na wydzieleniu jej



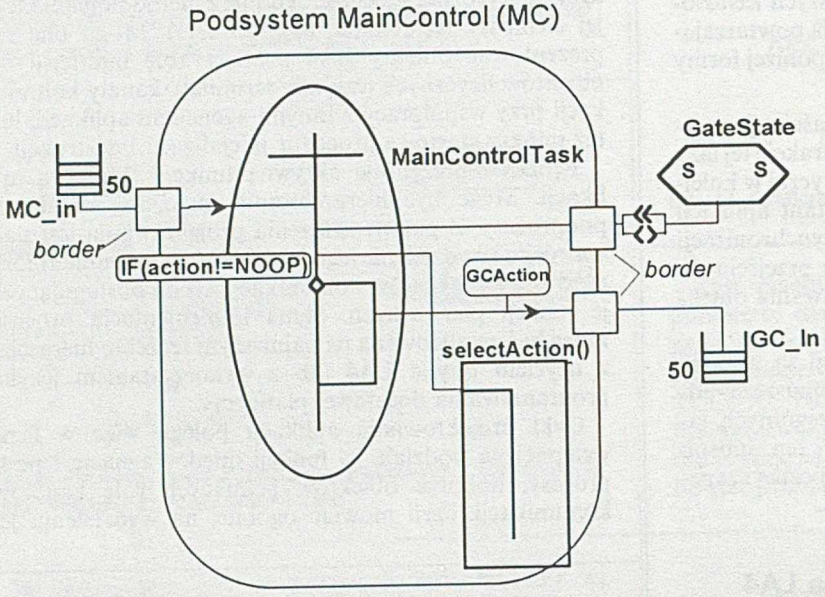
Rys. 13. Strefa ruchu wahadłowego

Rys. 14. Dekompozycja aplikacji, przepływ informacji między podsystemami

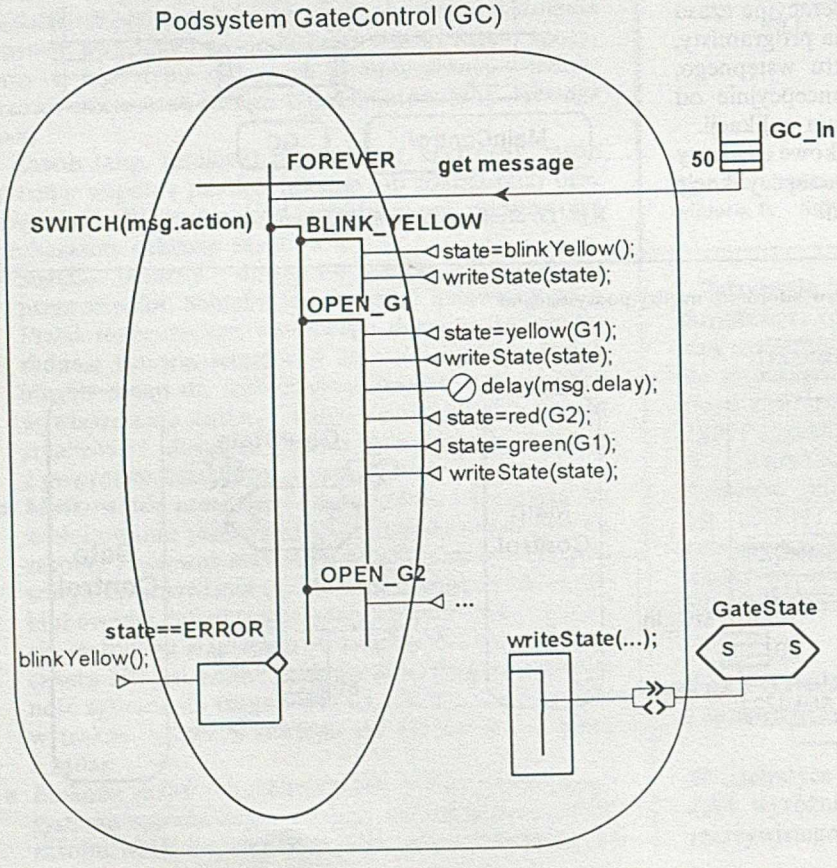


samodzielnych elementów składowych i ustaleniu interfejsu komunikacji i synchronizacji. W dalszej fazie uszczegółowiania projektu jest przeprowadzana dekompozycja procesów na dalsze podprocesy lub modelowanie bezpośrednio z użyciem standardowych elementów języka.
Na rys. 13 pokazano przykład aplikacji czasu rzeczywistego sterującej światłami ulicznymi. Droga dwukierunkowa zwęża się uniemożliwiając mijanie się pojazdów (odcinek *G2–G1*). Ruch odbywa się w tym miejscu w sposób wahadłowy: na przemian są przepuszczane grupy pojazdów z lewej i prawej strony. Przed wjazdem do strefy ruchu wahadłowego (strefy krytycznej) są ustawione światła (bramki *G1* i *G2*). U wlotu i wylotu ze strefy ruchu wahadłowego zainstalowano dwa czujniki: *s2* i *s3* pozwalające wykrywać przejeżdżające pojazdy. Zadaniem podsystemu CriticalZone (CZ) jest śledzenie stanu strefy, tak aby uniknąć zmiany świateł w momencie, kiedy nie jest ona pusta. W pewnej odległości od krańców strefy zainstalowano czujniki *s1* i *s2* rejestrujące zbliżające się pojazdy. Informacja ta jest wykorzystywana do poprawienia płynności ruchu w warunkach małego natężenia (zmniejszenie czasu oczekiwania na zmianę świateł). Podsystemy InputZone1 (IZ1) i InputZone2 (IZ2) rejestrują liczbę oczekujących pojazdów i czas oczekiwania.

Rys. 15. Implementacja podsystemu Main Control



Rys. 16. Implementacja podsystemu Gate Control



Przykład aplikacji

Działanie aplikacji jest koordynowane przez podsystem sterownika głównego MainControl (MC). Podstawowym jego zadaniem jest analiza stanu strefy ruchu wahadłowego i przesyłanie decyzji do podsystemu wykonawczego sterownika bramki GateControl (GC).

Na rys. 14 została pokazana struktura aplikacji na najwyższym poziomie hierarchii. Dla uproszczenia nie zaznaczono podsystemu realizującego fazy kreacji i destrukcji. Przepływ informacji odbywa się od czujników *s1, s2, s3, s4*. Zdarzenia fizyczne, za pośrednictwem procedur obsługi przerwań, trafiają do podsystemów InputZone1, InputZone2 i CriticalZone. Podsystemy te uaktualniają swoje wewnętrzne wektory stanu określające liczbę pojazdów w kontrolowanych strefach i czasy oczekiwania.

Stany podsystemów wejściowych są przesyłane, jako odpowiednie zdarzenia logiczne, do podsystemu MainControl za pośrednictwem kolejki komunikatów MC_In, pełniącej rolę kanału komunikacji. Możliwa jest tu także sygnalizacja sytuacji wyjątkowych, np. przekroczenie maksymalnego czasu oczekiwania, pojawienie się błędów śledzenia.

Podsystem MainControl – jest tu realizowana główna pętla decyzyjna aplikacji. Podsystem czeka na zdarzenia logiczne generowane przez moduły IZ1, IZ2 i CZ (rys. 15). Po otrzymaniu informacji o zmianie stanu kont-

rolowanych przez nie stref (np. opróżnienie się strefy ruchu wahadłowego) sprawdza stan świateł (strukturę dzielonej pamięci GateState uaktualnianą przez moduł GC), podejmuje decyzję, czy powinna nastąpić zmiana (wywołanie procedury selectAction ()) i przekazuje decyzję do podsystemu GC, za pośrednictwem komunikatu wysyłanego do wejściowej kolejki komunikatów GC_In.

Podsystem GateControl pełni w aplikacji rolę wykonawczą – realizuje bezpośrednią obsługę sprzętu. Podsystem obejmuje grupę procedur, takich jak **yellow**() (zmień światła na żółte), **red**() (zmień światła na czerwone dla danej bramki), itp. Wynik każdej operacji jest zapisywany w lokalnej zmiennej **state**, a następnie za każdym razem umieszczany w pamięci dzielonej GateSate. Zadanie realizujące główną pętlę (rys. 16) czeka na polecenia przekazane do kolejki wejściowej GC_In w zależności od wymaganej akcji (jak OPEN_G1) woła kolejno kilka funkcji (jak **yellow**(), **delay**(), **red**(G2), **green**(G1)).

* * *

W artykule przedstawiono język LA4 przeznaczony do specyfikowania oprogramowania czasu rzeczywistego. Główny nacisk położono na prezentację podstawowych elementów języka oraz pokazanie możliwości jego zastosowania do specyfikacji i projektowania aplikacji czasu rzeczywistego. Obiekty języka LA4 modelują podstawowe mechanizmy systemu operacyjnego czasu rzeczywistego, dzięki czemu mogą być wykorzystane do projektowania struktury aplikacji: dekompozycja na procesy-zadania, określenie mechanizmów synchronizacji i wymiany informacji. Własność ta umożliwia ponadto automatyczne generowanie „ram" aplikacji na podstawie tekstowej specyfikacji w języku LA4_T (plik ASCII). Ramy te mogą być wypełniane przez „wstrzykiwanie" kodu funkcji w języku docelowym [9]. Odpowiednie prace dla systemu operacyjnego iRMX są prowadzone w INSA Lyon i dla systemu VxWorks w Katedrze Automatyki AGH. Rozszerzenia języka o *proces* i *agencję* dostarczają narzędzi do budowania hierarchicznych specyfikacji zgodnie z metodyką projektowania strukturalnego. Aplikacja specyfikowana z użyciem tego języka może być również analizowana pod kątem wymogów poprawnościowych [7, 8]. Analiza ta może być prowadzona na różnych poziomach abstrakcji i pod kątem spełnienia rozmaitych wymagań [7].

Praca została wykonana w ramach grantu KBN 8T11A 01308: *Metody formalne i narzędzia wspomagające projektowanie i analizę oprogramowania czasu rzeczywistego do sterowania procesami.*

LITERATURA

[1] IEEE/ANSI Standard: Guide for Software Quality Assurance Planning, IEEE, 1986
[2] Parkinson J.: Making CASE Work, NCC Blakwell Ltd, (2) 1991
[3] Schwarz J-J., Skubich J., Miguel M.: A Graphical Language for Multitasking Real-Time Application Design. An Application to iRMX Programming, Proceedings of the 7th Int. Conf. IRUG, Baltimore, USA, (1–7) 1991
[4] Schwarz J-J.: Language d'Aide a la Conception d'Applications Multitaches Temps Reel – APII-AFCET, 1992, Dunod, Vol 26, No 5, pp. 355–385
[5] Schwarz J-J., Skubich J-J., Szwed P., Maranzana M.: Real Time Multitasking Design with a Graphical Tool – First IEEE Workshop on Real Time Applications p. 33–36, New York 1993
[6] Schwarz J-J., Skubich J-J., Szwed P., Maranzana M.: CASE Tools for iRMX Applications. 8th IRUG International Conference Proceedings pp. 43–49. Portland, Oregon, USA. October 1993
[7] Szmuc T., Szwed P.: Towards Automatic Correctness Verification of Real-Time Programs, Applied Mathematics and Computer Science, Vol. 4, No. 4, pp. 643–660, 1994
[8] Szmuc T., Szwed P., Schwarz J-J., Skubich J-J.: Hierarchical Correctness Verification in Multiphase Real-Time Software Design, Proceedings of the 17th Workshop on Real-Time Programming, Lake Constance, Germany, Pergamon Press, pp. 87–94, June 1994
[9] Szwed P.: LACATRE Reference Guide, Opracowanie Instytutu Automatyki, Akademii Górniczo-Hutniczej, Kraków, 1994
[10] Ward P., Mellor S.: Structured Development for Real-Time Systems, Prentice Hall, 1985.

Centrum Informatyczne Politechniki Wrocławskiej
oraz Instytut Automatyki Systemów Energetycznych
zapraszają na III Konferencję

Systemy czasu rzeczywistego '96

organizowaną pod hasłem:
Wytwarzanie oprogramowania do zastosowań mających wpływ na bezpieczeństwo

Konferencja odbędzie się w dniach **10–13 września 1996 r. w Szklarskiej Porębie** w Ośrodku szkoleniowo-wypoczynkowym Politechniki Wrocławskiej.

Zakres tematyczny konferencji:
- Metody i narzędzia analizy i modelowania systemów czasu rzeczywistego
- Bezpieczeństwo i niezawodność oprogramowania
- Metody i narzędzia specyfikacji, projektowania i implementacji oprogramowania
- Ocena wydajności oprogramowania
- Normalizacja wytwarzania i oceny jakości oprogramowania, atestacja oprogramowania
- Zastosowania przemysłowe: systemy komunikacji w czasie rzeczywistym, bazy danych czasu rzeczywistego, systemy doradcze, wbudowane systemy sterowania, symulacja i animacja, multimedia.

Organizacja konferencji:
10 września – wykłady szkoleniowe. Konspekty proponowanych wykładów należy nadsyłać w terminie do dnia 31 marca br.
11–13 września – prezentacja referatów omawiających wyniki aktualnie prowadzonych prac.
Przewiduje się wygłaszanie referatów przez zaproszonych specjalistów.

Zgłaszanie referatów:
Pełne teksty roboczej wersji referatów o objętości do 10 stron, należy przesyłać do dnia 31 marca br. Po ich akceptacji autorzy powinni przesłać teksty ostatecznej wersji referatów (napisanych w edytorze Word for Windows 6.0), które będą włączone do programu konferencji. Konferencja będzie się odbywać w języku polskim.

Adresy do korespondencji:
Prof. Zygmunt Mazur
Politechnika Wrocławska, Centrum Informatyczne
Wybrzeże Wyspiańskiego 27,
50-370 Wrocław
Sekretariat CI PWr.
– **tel.: (0-71) 20-35-16, 22-57**
– **faks: (0-71) 21-10-18**

Sekretariat konferencji:
e-mail: konf2@pwr.wroc.pl
Prof. Zbigniew Huzar:
zhuzar@ci-2.ci.pwr.pl
Prof. Zygmunt Mazur:
mazur@ci-2.ci.pwr.pl

I Konferencja „QNX in Real World"

Pierwsza Międzynarodowa Konferencja „QNX in Real World", poświęcona systemowi operacyjnemu QNX i jego zastosowaniom w kraju, odbyła się w Warszawie w połowie stycznia br. Konferencja stała się również okazją do promocji książki „QNX – system operacyjny" Krzysztofa Sachy. Impreza została zorganizowana przez Polski Komitet CAMAC i firmę X-Serwis, integratora systemu QNX, przy współudziale Instytutu Systemów Informatycznych WAT i Instytutu Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Wydarzenie spotkało się z dużym zainteresowaniem środowiska, oczekiwano około trzystu gości, zjawiło się prawie o połowę więcej.

Pierwszym wątkiem konferencji była ogólna prezentacja systemu. QNX – produkt kanadyjskiej firmy QSSL (QNX *Software Systems, Limited*) rozwija się już od wczesnych lat osiemdziesiątych. Jest to oparty na mikrokernelu modularny, zorientowany sieciowo, wielozadaniowy, wielodostępny system operacyjny czasu rzeczywistego dla mikroprocesorów rodziny 80×86, zgodny ze standardem POSIX. Duża efektywność w systemach rozproszonych, skalowalność i małe wymagania sprzętowe (jak dotąd żaden z kerneli QNX nie zajmował powyżej 10KB) sprawiają, iż znajduje szerokie zastosowanie, od niewielkich systemów wbudowanych, po duże sieciowe serwery baz danych, w aparaturze medycznej i pomiarowej, systemach kontroli i sterowania, a nawet w czołgach i na okrętach wojennych.

Zakres zastosowań systemu QNX w Polsce stale rośnie i temu właśnie poświęcony był drugi wątek konferencji. Obecnie, wg szacunkowych danych firmy X-Serwis, system ma już ponad półtora tysiąca użytkowników, łącznie z klientami konkurencji. Do tego należy doliczyć import bezpośrednio z zagranicy oraz bliżej nie ustaloną liczbę tych, którzy sami nawet nie wiedzą, że używają QNX, gdyż nabyli go razem z aparaturą. Jak dotąd sprzedano stosunkowo dużo pakietów dla programistów, co świadczy o ufności polskich firm programistycznych w chłonność rynku. Przewiduje się, że wraz z wdrażaniem rodzimego oprogramowania, które jakością bez wątpienia nie ustępuje zagranicznemu, będąc przy tym tańszym, dostosowanym do warunków lokalnych, i mającym dobry serwis techniczny, zapotrzebowanie na podstawowe pakiety systemu operacyjnego QNX będzie lawinowo wzrastać. Prelegenci przedstawili wdrożone przez siebie, faktycznie działające w kraju aplikacje z zastosowaniem QNX. Krótkie referaty obejmowały co ciekawsze zastosowania w takich działach, jak przemysł, telekomunikacja, wojsko, handel, medycyna i nauka.

Kolejnym elementem konferencji było omówienie perspektyw rozwoju QNX. Przedstawiciel QSSL z Kanady zapowiedział bliskie już nadejście kolejnej wersji systemu QNX-4.3. Z tą właśnie wersją system stanie się w pełni 32-bitowy, choć można będzie nadal korzystać z 16-bitowych binariów. Nowy menadżer procesów, Proc32, pozwoli na jednoczesne uruchomienie do 2048 procesów. Czas przełączania zadań ma zmniejszyć się o 30%, a szybkość komunikacji międzyprocesowej wzrosnąć o 10%. Wybitnie zwiększona ma być sprawność menadżera plików, Fsys. Network Manager uzyskał możliwości automatycznego łączenia (ang. *bridging*) pomiędzy różnymi sieciami fizycznymi. Z myślą o użytkownikach komunikujących się z QNX za pośrednictwem systemu Windows uaktualniono Dosfs tak, aby można było używać długich nazw zbiorów Windows 95. Photon, specjalny interfejs graficzny dla QNX (mały, lecz o dużych możliwościach), będzie standardowo dołączony do pakietu.

Konferencji towarzyszyła niewielka wystawa, na której zaprezentowało się osiem firm dostarczających sprzęt i akcesoria powszechne w zastosowaniach QNX: kable i karty sieciowe, komputery w pełnej gamie form i obudów, karty i moduły AD i DA, urządzenia pomiarowe, oprogramowanie oraz pełne rozwiązania sprzętowo-programistyczne.

Konferencja miała charakter popularyzatorski. Organizatorzy, którzy jak sami przyznają, wystąpili po raz pierwszy w roli gospodarzy, zapowiedzieli, że konferencja QNX odbywać się będzie co roku i wchodzi na stałe do kalendarium światowych imprez poświęconych temu systemowi operacyjnemu.

Przemysław Baszkiewicz

QNX
w księgarniach...

W ubiegłym roku ukazały się dwie książki dotyczące QNX. Obie zostały wydane przez firmy, które współzawodniczą na rynku sprzedaży systemu QNX i obie najprawdopodobniej znajdują się w księgarniach obok siebie.

„System operacyjny QNX 4.2" Wojciecha Lercha, wydany przez firmę Quantum z Wrocławia można by, jak sądzę, określić mianem pozycji dla każdego: dla osób, które nigdy się z systemami typu UNIXa nie zetknęły i dla użytkowników zaawansowanych. „QNX – system operacyjny" Krzysztofa Sachy, wydany przez warszawską firmę X-Serwis jest adresowany raczej do czytelników średnio i bardziej zaawansowanych. Mocniej koncentruje się na cechach specyficznych systemu QNX i posługuje się językiem bardziej sformalizowanym. Obydwie książki są ciekawe i lektura jednej nie wyklucza przeczytania z równym zainteresowaniem drugiej.

Oprócz samej treści ważne jest także opracowanie książek. Obydwie książki opatrzone są bibliografią, jednak w książce Sachy jest ona nieco bogatsza. Książka Sachy ma również bardzo wygodne w użyciu zestawienie poleceń systemowych według zastosowania. Książka Lercha zawiera z kolei indeks haseł, którego niestety brak jest u Sachy. Również u Lercha znalazł się obszerny spis firm światowych oferujących sprzęt i oprogramowanie dla QNX. Sporym natomiast potknięciem, jeśli można to tak delikatnie ująć, jest w tej pozycji spis firm zajmujących się QNX w Polsce: pominięto milczeniem istnienie konkurencyjnej firmy X-Serwis oraz wielu producentów oprogramowania.

Obie pozycje prezentują QNX odrobinę inaczej i sądzę, że jest kwestią indywidualnego gustu, którą z nich czytelnik wybierze. Można oczywiście kupić obie. **(pb)**

Wojciech Lerch, System operacyjny QNX wersja 4.2. Wrocław: Quantum, 1995, 156 s.
Krzysztof Sacha, QNX – system operacyjny. Warszawa: X-Serwis, 1995, 142 s.

QNX w Internecie

Zasoby QNX w Internecie są niewątpliwie skromne, w porównaniu z ogromem materiałów dla najpopularniejszych systemów klasy UNIX, z pewnością jednak użytkownicy QNX nie mają powodu czuć się pokrzywdzeni. Już poniższy, bardzo pobieżny przegląd, obejmujący jedynie miejsca typowo QNX-owe, od których najłatwiej jest rozpocząć poszukiwania w sieci, daje pewne pojęcie, że zasobów tych nie jest wcale mało.

Dobrym punktem startowym jest strona **http://www.qnx.com** firmy QSSL, producenta systemu. Zebrano na niej najważniejsze informacje o samym QNX, jego producencie, dotyczących go nowościach oraz wiele przykładów zastosowania systemu. Przydatna jest bibliografia aktualnych publikacji na temat QNX (dokument **/pr/qnxinp.html**). Serwer QSSL oferuje również usługę ftp (**ftp://ftp.qnx.com**), można z niego ściągnąć rozmaite dokumenty dotyczące systemu, darmowe i demonstracyjne oprogramowanie, uzupełnienia do pakietów komercyjnych i inne materiały. Wsparcie techniczne można uzyskać przez e-mail (**info@qnx.com**)lub wchodząc przez WWW do systemu konferencyjnego QUICS, i zadając pytania ekspertom. Tym, którzy wybiorą z dowolnej strony WWW odsyłacz (ang. *link*) „Info Request Form" i wypełnią ankietę na temat swoich zainteresowań QNX'owych, firma obiecuje przesłać przewodnik „QNX System Architecture".

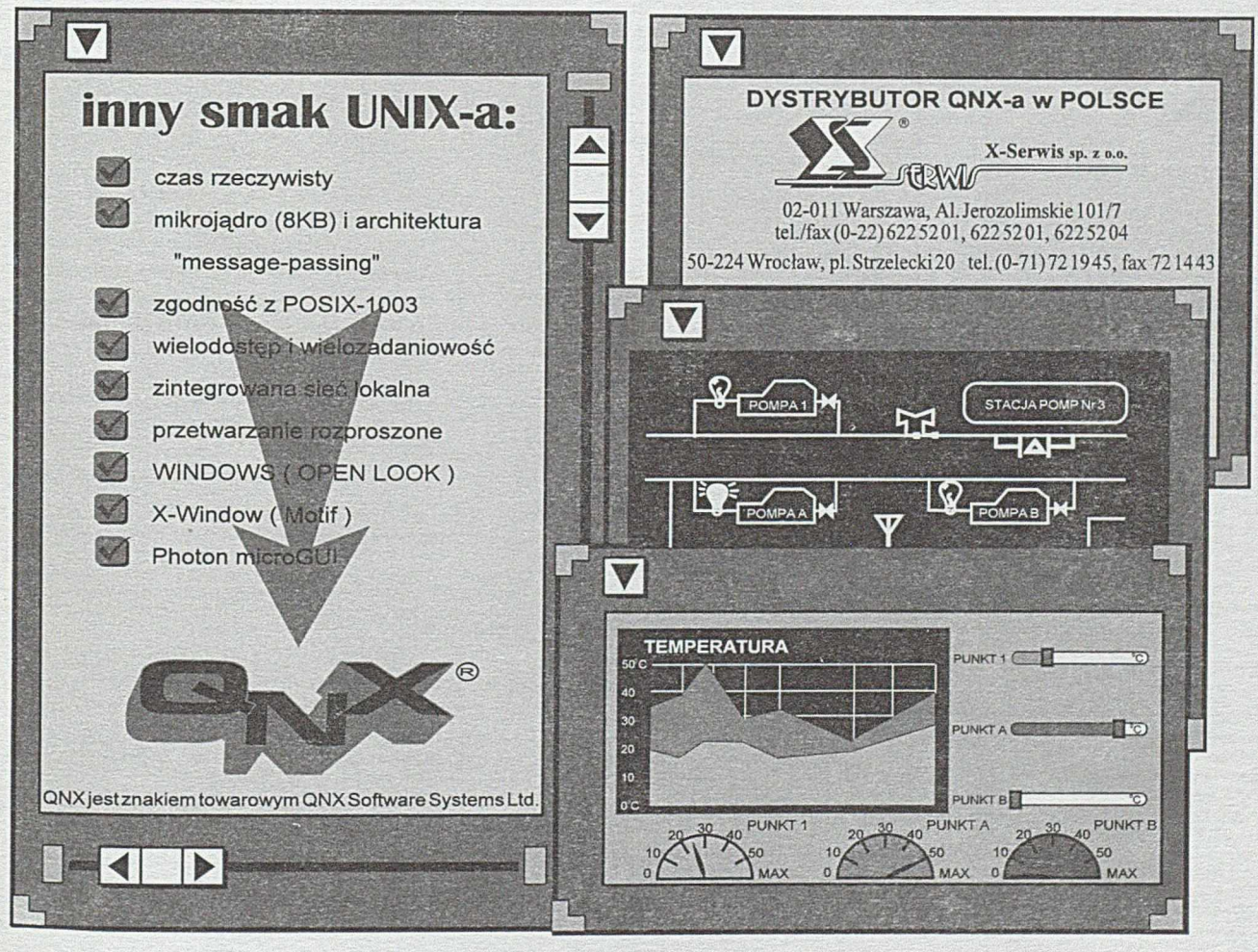
Dalszą eksplorację Internetu można prowadzić korzystając z różnych spisów odsyłaczy o tematyce QNX. Jeden z nich znajduje się w omówionym już **www.qnx.com** i zbiera adresy WWW firm zajmujących się oprogramowaniem i sprzętem dla QNX'a (odsyłacz „Third Party Products & Services"). Dobry spis odsyłaczy znaleźć można na **http://www.passageway.com/camz/qnx/** (C.A.M. Zimmerman). Do spisu tego można zgłosić własne WWW wypełniając na ekranie odpowiedni formularz. Inne warte odwiedzenia spisy odsyłaczy to:
http://www.hookup.net/~fcsoft/otherplayers.html (FC Software) i
http://www.owt.com/users/chubbard/html/qnx.html (C. Hubbard). Chociaż zawarte na wymienionych stronach zestawy odsyłaczy odwołują się w dużym stopniu do tych samych adresów WWW, to jednak każda ze stron jest na swój sposób godna zainteresowania, dzięki różnemu ujęciu tematyki.

Poszukujący dodatkowych informacji z pewnością chętnie skorzystają z comp.os.qnx, usenet'owej grupy dyskusyjnej użytkowników QNX z całego świata – jest ona powszechnie dostępna w sieci (np. na **news.mimuw.edu.pl**). Można tu uzyskać odpowiedź na niemal każde pytanie na temat QNX. Bardziej widocznym efektem dyskusji tej grupy jest QNX 4.xx FAQ, czyli dokument podsumowujący najczęściej pojawiające się w comp.os.qnx pytania i odpowiedzi na nie. Dokument jest dostępny przez **ftp://www.qnx.com/usr/free/FAQ** lub na wspomnianym już wcześniej **www.passageway.com** w formie hipertekstu.

Obraz zasobów sieciowych oczywiście byłby niepełny bez informacji, iż w sieci jest dostępna QNX'owa kompilacja gry Doom (**ftp://ftp.qnx.com/usr/free/doom/**), bez czego żaden system operacyjny obejść się nie może.

W Polsce znalezienie się w WWW planuje firma X-Serwis (**qnx@ikp.atm.waw.pl**), dystrybutor QNX, co ma ułatwić techniczną obsługę klientów. Na początek, prawdopodobnie w ciągu miesiąca lub dwóch ma powstać BBS, a dopiero w dalszej, niestety bliżej nieokreślonej, przyszłości własna strona firmy.

(pb)



informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

Uwaga, dzięki gościnności Polska Online ukazuje się już elektroniczna wersja naszego miesięcznika. Oto nasz adres w Internecie:

http:/www.pol.pl/news._p/informa.htm

nr 2 1996
Luty
Rok wydania XXXII

REDAGUJE ZESPÓŁ:
mgr Krystyna **KARWICKA-RYCHLEWICZ** (redaktor naczelny)
mgr Jerzy **SZYLLER** (z-ca redaktora naczelnego)
mgr Teresa **JABŁOŃSKA** (sekretarz redakcji)
Przemysław **BASZKIEWICZ** (redaktor)
mgr Zdzisław **ŻURAKOWSKI** (redaktor działu)
Alina **KLEPACZ** (sekretariat)

KOLEGIUM REDAKCYJNE:
prof. dr hab. Leonard BOLC,
mgr inż. Piotr FUGLEWICZ,
prof. dr hab. Jan GOLIŃSKI,
dr inż. Zenon KULPA,
prof. dr inż. Jan MULAWKA,
prof. dr hab. Wojciech OLEJNICZAK,
mgr inż. Jan RYŻKO,
dr Witold STANISZKIS,
dr inż. Jacek STOCHLAK,
prof. dr hab. Maciej STOLARSKI,
prof. dr hab. Zdzisław SZYJEWSKI,
prof. dr hab. inż. Ryszard TADEUSIEWICZ,
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628 i 644
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail:
informat@pol.pl

Materiałów nie zamówionych redakcja nie zwraca

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach $3^1/_2$" – w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6 – oraz odbitki na papierze. Teksty nie mogą przekraczać 20 tys. znaków.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- Redakcja
- Agencja reklamowa KUBA
  ul. Kopernika 28 m 23
  00-336 Warszawa
  tel.: 27-98-66, faks: 27-24-91

lub
- Dział Reklamy i Marketingu
  00-950 Warszawa, ul. Mazowiecka 12
  tel.: 27-43-66, faks: 26-80-16

W numerze:

	Strona
Integracja i partnerstwo – *Jerzy Szyller*	I okł.
informacje	1–8
publikacje	
ARIS-Toolset – narzędzia dla Business Process Reengineering – *Mirosława Lasek*	9
Zagadnienia pozyskiwania wiedzy dla sterowników rozmytych (1) – *Jarosław Stańczak*	14
Integracja zarządzania sieciami IPX/SPX i TCP/IP (2) – *Tomasz Lis, Artur Sobkowiak, Maciej Stroiński*	18
Programowanie w języku systemów eksperckich CLIPS (4). Integracja systemów eksperckich z programami opracowanymi w innym środowisku – *Tomasz Pytlak*	22
Systemy czasu rzeczywistego	
Metody analizy i specyfikacji wymagań. Strukturalno-graficzny język opisu i projektowania aplikacji czasu rzeczywistego – *Tomasz Szmuc, Piotr Szwed*	26
III Konferencja „Systemy czasu rzeczywistego '96"	33
I Konferencja „QNX in Real World" – *Przemysław Baszkiewicz*	34

W najbliższych numerach:

W marcowym numerze INFORMATYKI zostaną opublikowane materiały z konferencji „Język i Technologia '95", a wśród nich:

- **Jan Roukens w artykule programowym odzwierciedla dążenie ku wielojęzycznemu społeczeństwu informacji w Europie.**
- **Antonio Zampolli charakteryzuje współpracę międzynarodową w dziedzinie zasobów języka naturalnego.**
- **Witold Abramowicz charakteryzuje retorykę hipertekstu.**
- **Leonard Bolc i in. przedstawiają badania związane z opracowaniem metod opisu składni języków naturalnych.**
- **Czesław Basztura przedstawia podstawowe zagadnienia głosowej komunikacji człowiek-komputer.**
- **Dafydd Gibbon omawia technologie i zasoby języka mówionego i pisanego.**
- **Bente Maegaard i Karsten Strorup zajmują się tłumaczeniem maszynowym.**
- **Zygmunt Vetulani podaje założenia metodologiczne rozumienia języka polskiego przez komputer.**

Warunki prenumeraty na 1996 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-716 Warszawa, skr. poczt. 1004
konto:
PBK S.A. III 0/Warszawa nr 370015-1573-139-11

Wpłaty na prenumeratę przyjmują także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 5,00 zł (50 000 zł), ulgowa 3,75 zł (37 500 zł)
Wartość prenumeraty w zł:
Normalna: kwartalna 15,00 zł (150 000 zł), półroczna 30,00 zł (300 000 zł), roczna 60,00 zł (600 000 zł)
Ulgowa: kwartalna 11,25 zł (112 500 zł), półroczna 22,50 zł (225 000 zł), roczna 45,00 zł (450 000 zł)

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 5/96

Rodan-System Sp. z o.o.

IBM Polska

zapraszają wszystkich chętnych do udziału w

III

MIĘDZYNARODOWYCH

ZAWODACH NARCIARSKICH

O PUCHAR Rodan-System

Zawody odbędą się **16 marca 1996r. na stoku Nosal w Zakopanem.**

PROGRAM IMPREZY

10^{00} - 10^{30} - Rejestracja uczestników seminarium, połączona z rejestracją startujących o Puchar Rodan System zawodników, Hotel Kasprowy;

SEMINARIUM - „Biuro przyszłości" sala konferencyjna Hotelu Kasprowy:

10^{30} - 11^{15} - Wykorzystanie nowoczesnych środowisk automatyzacji prac biurowych w dziedzinie reinżynierii procesów działalności,
prowadzący: **Witold Staniszkis Rodan System;**

11^{15} - 12^{00} - Architektura systemu zarządzania dokumentami i przepływem prac,
prowadzący: **Bartosz Kiepuszewski Rodan System;**

12^{00} - 12^{30} - przerwa / zimny bufet / w kuluarach - pokaz systemu zarządzania dokumentami i przepływem prac, zrealizowanego w firmie Rodan System;

12^{30} - 13^{15} - System IBM FlowMark - architektura i zastosowania,
prowadzący: **Tomislav Begovac, IBM;**

13^{15} - 14^{00} - Zarządzanie hierarchiczną pamięcią dokumentów
prowadzący: **Marcelo Nahum-Feber - Rodan System;**

ZAWODY O PUCHAR Rodan-System - stok Nosal

15^{30} - 16^{00} - rejestracja zawodników

16^{00} - 18^{00} - 2 przejazdy (slalom gigant) w dwóch grupach wiekowych: - powyżej 22 lat (juniorzy),
- do 22 lat (super-juniorzy),

18^{00} zakończenie zawodów na stoku Nosal

20^{00} uroczyste rozdanie nagród i bankiet w szałasie przy końcu ul. Piłsudskiego na przeciwko TOPR.

Opłata startowa wynosi 40 zł od osoby.
Koszty dojazdu i zakwaterowania uczestnicy pokrywają we własnym zakresie.
Zgłoszenia wraz z kopią przelewu prosimy nadsyłać do dnia 13 marca 1996r.

Rodan-System sp.z o.o. ul. Jagielska 50c, 02-886 Warszawa, Tel. 643 92 08 Fax 643 92 10
numer konta: PKO SA VII O/Warszawa, nr 501190-10003978-2511-3-1110.
Informacji udziela i zgłoszenia przyjmuje pani Agnieszka Rowicka.

0/296

Microsoft® Office

Standard

dla Windows® 95

Wszystko co potrzebne w Twoim biurze

Microsoft Office dla Windows 95 to pakiet zintegrowany najnowszej generacji, teraz dostępny także w polskiej wersji.
Microsoft Office dla Windows 95 wersja Standard zawiera: arkusz kalkulacyjny Microsoft Excel, edytor tekstów Microsoft Word, program do tworzenia prezentacji Microsoft PowerPoint i elektroniczny kalendarz/organizator Microsoft Schedule+. Microsoft Office wykorzystuje w pełni mechanizmy systemu operacyjnego Microsoft Windows 95, wszystkie aplikacje obsługują długie nazwy plików, skróty i są wielozadaniowe. Użytkownik może bezpiecznie przełączać się pomiędzy aplikacjami w 32-bitowym środowisku. Za pomocą Microsoft Exchange można z jednego miejsca wysyłać faksy i pocztę elektroniczną. Dzięki technologii OfficeLinks można łączyć różnego rodzaju pliki: listy, raporty, arkusze czy bazy danych w wygodną formę Spinacza i przechowywać je jako jeden obiekt.
Krótko mówiąc: Microsoft Office pozwala Ci skoncentrować się na najważniejszym zadaniu - Twojej pracy.

Microsoft®

COMPUTER 2000

Microsoft, Windows są zastrzeżonymi znakami towarowymi firmy Microsoft Corporation